NASA Contractor Report 158993

MODELS AND TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS

Semi-Annual Status Report Number 4

(NASA-CR-158993) MODELS AND TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS Semiannual Status Report, 1 Nov. 1977 - 31 May 1978 (Michigan Univ.) 153 p HC A08/MF A01 CSCL 09B G3/60
N79-17564
Unclas
43727

John F. Meyer

THE UNIVERSITY OF MICHIGAN
Ann Arbor, Michigan 48109

NASA Grant NSG-1306
July 1978

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

MODELS AND TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS

Semi-Annual Status Report Number 4
Covering the Reporting Period
1 November 1977 - 31 May 1978

Prepared For
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23365
Sal J. Bavuso, NASA Technical Officer
NASA Grant NSG 1306

John F. Meyer, Principal Investigator
Department of Electrical and Computer Engineering
Systems Engineering Laboratory
SEL Report Number 120
The University of Michigan
Ann Arbor, Michigan 48109

July 1978

TABLE OF CONTENTS

1. INTRODUCTION

This report is the fourth Semi-Annual Status Report on the research project "Models and Techniques for Evaluating the Effectiveness of Aircraft Computing Systems" being conducted for the NASA Langley Research Center under NASA Grant 1306. The subject grant was initiated 1 May 1976 for a one year period and extended 1 May 1977 for a second one year period. This report concerns work accomplished during the second half of the second year (which included a one month no-cost extension at the end of the year), that is, the period from 1 November 1977 to 31 May 1978, hereafter referred to as the reporting period.

The purpose of this research project is to develop models, measures, and techniques for evaluating the effectiveness of aircraft computing systems. By "effectiveness" in this context we mean the extent to which the user, i.e., a commercial air carrier, may expect to benefit from the computational tasks accomplished by a computing system in the environment of an advanced commercial aircraft. Thus, the concept of effectiveness involves aspects of system performance, relaibility, and worth (value, benefit) which must be appropriately integrated in the process of evaluating system effectiveness. Specifically, the primary objectives of this project are:

(1) The development of system models that can provide a basis for the formulation and evaluation of aircraft computer system effectiveness,

(2) The formulation of quantitative measures of system effectiveness, and

(3) The development of analytic and simulation techniques for evaluating the effectiveness of a proposed or existing aircraft computer.

Effort during the reporting period has also been devoted to the documentation of research results for dissemination at technical conferences and in the open literature. In particular, some definitive results of the first year's activity were submitted for presentation at the 8th International Symposium on Fault-Tolerant Computing (Toulouse, France, June 21-23, 1978). This paper was accepted and will appear in the Proceedings of FTCS-8 [4]. Another paper, based on the same work but stressing the unification of performance and reliability, has been accepted for presentation at the Symposium on Modelling and Simulation Methodology (Rehovot, Israel, August 13-18, 1978). More recent results concerning "functional dependence" (R-dependence) and its implications (see [3] and Section 3.1 of this report) have been presented at the 1978 Johns Hopkins Conference on Information Sciences and Systems (Baltimore, Maryland, March 1-3, 1978) and will be published in the Proceedings of that conference [5]. In addition, a paper focusing on the "performability evaluation of fault-tolerant multiprocessors" in a commercial aircraft environment has been accepted for presentation at the 1978 Government Microcircuit Applications Conference (Monterey, California, November 14-16, 1978). Finally, a slightly expanded version of the FTCS-8 paper [4] has been submitted for publication in the IEEE Transactions on Computers.

Section 2 of this report describes the manpower effort proposed for the current year, the personnel involved in conducting the investigation, and their levels of effort during the reporting period. Section 3, the body of the report, describes the technical status of the research performed during the reporting period.

3. TECHNICAL STATUS

The following is a comprehensive description of the research performed during the reporting period. The report is divided into three major sections under the headings:

3.1 Functional Dependence,

3.2 Evaluation Algorithms and Programs (METAPHOR),

3.3 Performability Evaluation of the SIFT Computer.

Section 3.1 describes our further investigation of the concept of "functional dependence" (R-dependence) and its relation to "structure-based" capability functions. The results of this investigation include some basic theorems characterizing R-dependence and R-dependent sets when the index set D is countably infinite. (Similar results obtained during the previous reporting period [3] assumed a finite index set.) Of more practical signifigance, however, is the use of these basic theorems to establish the fundamental limitations of reliability modeling that is based on "structure functions" or, equivalently, their representation by "fault-trees". In particular, it is shown (Theorem 6) that any phased system model, wherein the capability function can be described by a sequence of structure functions (fault-trees), is characterized by a total absence of R-dependence among the phases (where R is the set of all state trajectories corresponding to system "success"). One of the features of performability modeling, on the other hand, is its ability to accomodate interphase dependencies, as illustrated in the conclusion of Section 3.1.

Section 3.2 reviews our progress in the development of METAPHOR, a prototype software package to aid the evaluation of performability. This includes a discussion of the objectives and abilities of METAPHOR as well as a description of how the current implementation of the package is used. Effort has also been devoted to producing more detailed documentation of how METAPHOR is implemented. This documentation effort is still in progress, however, and will be completed during the next reporting period. A full report of this activity will be included in the next Semi-Annual Status Report.

Section 3.3 concerns the major part of our activity during the reporting period, a relatively comprehensive performability modeling and evaluation exercise involving the SIFT computer [8]. The computational environment is assumed to be a transoceanic flight of a commercial aircraft and the accomplishment set A is naturally defined in terms of attributes used by Ratner, et. al. [7] to distinguish the "criticalities" of various aircraft functions. The capability function γ_S of the "total system" (SIFT plus its environment) is described in terms of a 3-level model hierarchy, where each step of the modeling process is explained in considerable detail. Performability is then evaluated using the basic two-step computational procedure described previously (see [3], [4], for example), that is,

1) For each accomplishment level in A, determine the set of all state trajectories that result in a, that is, determine the inverse image $U_a = \gamma_S^{-1}(a)$,

2) Using the base model X_S, for each a in A, compute the probability of the trajectory set U_a (which is equal to the performability value $p_S(a)$).

Implementation of step 1) is described in more detail than it has been in the reports of previous evaluation exercises. In both steps, computations were aided by the current version of METAPHOR but many of the calculations, particularly in step 1), remain to be automated. This necessitated a great deal of tedious manual computation and resulted in computational errors that were difficult to locate. However, the results finally obtained appear to be correct since they satisfy several consistancy checks. More importantly, we believe that the work described in Section 3.3 comprises a significant step toward establishing the practicality of performability evaluation, particularly as it applies to aircraft computing systems.

3.1 Functional Dependence

During the reporting period, our investigation of the concept of "functional dependence" has continued with an emphasis on i) extending the theory to (countably) infinite coordinate sets and ii) using the theory to characterize the limitations of traditional structure-based reliability analysis. Although specific results of this effort have already been documented in a paper presented at the 1978 Johns Hopkins Conference (see [5]), the discussion that follows links the work of the current period to that described in previous reports and, thereby, serves to clarify the progress during the current reporting period.

3.1.1 Extension of the Definition of R-dependence

Prior to the reporting period, the investigation of functional dependence or, more formally, "R-dependence" has presumed that the underlying index set D (see [3], p. 28, Def. 1) is finite. Although not stated explicitly in Def. 1, the finiteness assumption becomes apparent in Def. 2 ([3], p.29) and is used in the proof of theorem 4 ([3], p.41). In our current applications of functional dependence, the index sets D are indeed finite since the indicies correspond to the decomposition of the state space into a finite number of component spaces and/or the decomposition of the utilization period into a finite number of phases. However, we anticipate applications where the user will be interested in long-run performability, in which case the utilization period may be unbounded. In such cases, assuming that each phase has finite duration, the number of phases will be countably infinite. To accommodate such situations, we have extended the definition of R-dependence to include infinite index sets and have reexamined the theory of R-dependence in the light of this extension.

To begin, the system coordinates (whether distinguished in time, space, or both) are represented by a countable set D, called the index set, where, as earlier, we assume that D is totally ordered relative to some underlying ordering relation. For example, if S is a system with n subsystem S_1, S_2, ..., S_n and the long-run behavior of the system is observed at discrete times t_1, t_2, t_3, ..., then

$$D = \{(i,j) \mid i \in \{1,2,\ldots,n\}, j \in \{1,2,3,\ldots\}\}$$

where (i,j) represents subsystem S_i observed at time t_j. D is then totally ordered in some convenient way, e.g., by the relation $\preceq$ where

$$(a,b) \preceq (c,d) \begin{array}{l} \text{if } a < c \\ \text{or } a = c \text{ and } b \leq d. \end{array}$$

Relative to D and some family of sets

$$\mathcal{Q} = \{Q_d \mid d \in D\}$$

indexed by D, the concept of R-dependence is based on the following types of sets.

Definition 1: A structured set (relative to D and $\mathcal{Q}$) is a subset of the Cartesian product of the sets in $\mathcal{Q}$, that is,

$$R \subseteq \mathop{\times}_{d \in D} Q_d$$

where the product is taken according to the ordering of D.

In the context of Definition 1, $d \in D$ is a coordinate and Q_d is the range of coordinate d. A set $C \subseteq D$ is called a coordinate set; the coordinates in C are subject to the ordering relation imposed on D.

When dealing with structured sets it is convenient to refer to the values taken on by a particular coordinate or set of coordinates. If $d \in D$, let $\xi_d : R \rightarrow Q_d$ denote the projection of R on d, that is,

$$\xi_d(r_1, \ldots, r_d, \ldots) = r_d.$$

Extending such projections to coordinate sets:

Definition 2: If $C \subseteq D$ is a coordinate set where $C = \{c_1, c_2, c_3, \ldots\}$ (c_1 is the first element of C according to the ordering of D, c_2 is the second element, etc.) the projection of R on C is the function

$$\xi_C : R \rightarrow \bigtimes_{c \in C} Q_c$$

where $\xi_C(r) = (\xi_{c_1}(r), \ \xi_{c_2}(r), \ \xi_{c_3}(r), \ldots)$. If $C = \phi$ (the empty set), $\xi_\phi : R \rightarrow \{1_\phi\}$ where 1_ϕ is an arbitrary constant.

For example, suppose $D = \{1,2,3\}$ with the natural ordering, and $Q_i = \{0,1\}$, $i \in D$. Then $\xi_{\{1,2\}}((0,1,1)) = (0,1)$, and $\xi_{\{3\}}((1,1,0)) = (0)$. When C is a singleton set $\{d\}$, $\xi_{\{d\}}$ will usually be denoted as ξ_d.

With the above preliminaries and with a slight notational change to eliminate some confusion that arose in the previous status report, R-Dependence is defined as follows.

Definition 3: If R is a structured set (relative to D and $\mathcal{D}$) and $A, B \subseteq D$ then A R-depends on B (denoted $A \, \Delta_R \, B$) if $\exists v \in \xi_A(R), \exists w \in \xi_B(R)$

such that $\forall r \in R$ [$\xi_B(r) = w$ implies $\xi_A(r) \neq v$]. A is <u>R-independent</u> of B ($A \not\Delta_R B$) if A does not R-depend on B.

The implications of the above definition, when so extended to permit a countably infinite index set D, are examined in the subsections that follow.

3.1.2 Basic Properties

Regarding Δ_R ("R-depends on") as a relation on the set of all subsets of the index set D (i.e., the "power set" of D), we note, first of all, that the global properties of Δ_R are preserved when D becomes countably infinite. In particular, as established earlier for finite D, we find that Δ_R is symmetric but generally neither reflexive nor transitive. The symmetry of Δ_R (i.e., $A \Delta_R B$ implies $B \Delta_R A$) follows immediately from Definition 3 for if v and w are such that [$\xi_B(r) = w$ implies $\xi_A(r) \neq v$] then [$\xi_A(r) = v$ implies $\xi_B(r) \neq w$]. Regarding reflexivity, if $C \subseteq D$ it follows that C R-depends on C if and only if $|\xi_C(R)| > 1$. (If $|\xi_C(R)| > 1$ any two distinct elements of $\xi_C(R)$ can serve as the v and w of Def. 3; if $|\xi_C(R)| = 1$, distinct elements u and v do not exist and, hence, C can not R-depend on C.)

Accordingly, Δ_R is generally not reflexive since there may exist a coordinate set C for which $|\xi_C(R)| = 1$, i.e., its projection is a constant. On the other hand, in the special case where no such coordinate sets exist, it follows that Δ_R is a reflexive relation. Finally, as demonstrated in the previous report using the structured set $R = \{(0,0,0), (0,0,1), (1,0,0), (1,1,1)\}$ (see [3], p.35), Δ_R is generally not a transitive relation. (This finite example suffices since the generalization to countable index sets D includes the special case where D is finite.)

The alternative characterizations of R-dependence (see [3], p. 34, theorem 1) also hold when D is countable, where we have found that the "partition characterization" (part iii) of Theorem 1) is especially useful. In the interest of clarity, this characterization will be restated using the notation of Def. 3, and then proved directly (as opposed to the earlier proof which involved two characterizations). The partition characterization is motivated by the fact that the "knowledge" or "information" conveyed by a coordinate set C can be regarded as classification of sequences in R, where two sequences are in the same class if they have the same projection on C. More precisely, if $C \subseteq D$ let $\equiv_C$ denote the "equivalence kernel" of ξ_C, i.e., for all $r, s \in R$, $r \equiv_C s$ iff $\xi_C(r) = \xi_C(s)$, and let π_C denote the partition of R induced by $\equiv_C$, that is, π_C is the set of all equivalence classes of $\equiv_C$. Finally, if $v \in \xi_C(R)$, let $\mathbb{B}_C(v)$ denote the "block" of π_C (equivalence class of $\equiv_C$) determined by v, that is, $\mathbb{B}_C(v) = \{r \in R \mid \xi_C(r) = v\}$. Then, in terms of these partitions, the concept of R-dependence can be characterized as follows.

<u>Theorem 1</u>: Let R be a structured set indexed by D, and let $A, B \subseteq D$. Then A R-depends on B if and only if $\exists v \in \xi_A(R)$, $\exists w \in \xi_B(R)$ such that $\mathbb{B}_A(v) \cap \mathbb{B}_B(w) = \phi$.

<u>Proof</u>: Suppose $A \Delta_R B$, and let v,w be as in Definition 3, i.e., $\forall r \in R[\xi_B(r) = w \Rightarrow \xi_A(r) \neq v]$. But $\xi_B(r) = w \Leftrightarrow r \in \mathbb{B}_B(w)$, and $\xi_A(r) \neq v \Leftrightarrow r \notin \mathbb{B}_A(v)$. Hence, $\forall r \in R[r \in \mathbb{B}_B(w) \Rightarrow r \notin \mathbb{B}_A(v)]$ which, in turn, implies that $\mathbb{B}_A(v) \cap \mathbb{B}_B(w) = \phi$. Conversely, suppose $\exists v \in \xi_A(R)$, $\exists w \in \xi_B(R)$

such that $\mathbb{B}_A(v) \cap \mathbb{B}_B(w) = \phi$. Then $\forall r \varepsilon R$, whenever $r \varepsilon \mathbb{B}_B(w)$, it must be the case that $r \not\varepsilon \mathbb{B}_A(v)$. Therefore, $\forall r \varepsilon R[\xi_B(r) = w \Rightarrow \xi_A(r) \neq v]$.

Theorem 1 thus provides a convenient algebraic characterization of R-dependence which, given partitions π_A and π_B, says that A R-depends on B if and only if there is a block in π_A and a block in π_B which have no elements in common.

Using Theorem 1, we can derive additional properties of R-dependence which are useful when searching for R-dependencies. As was the case for finite D (see[3], p.36, Lemma1) we observe, first of all, that if $A \Delta_R B$ then supersets of A must R-depend on supersets of B, that is:

<u>Theorem 2</u>: Let $A,B \subseteq D$. If $A \Delta_R B$ then, $\forall A' \supseteq A$ and $\forall B' \supseteq B$ such that $A',B' \subseteq D, \mid A' \Delta_R B'$.

<u>Proof</u>: Suppose $A \Delta_R B$ and let $A' \supseteq A$, $B' \supseteq B$. Then $\exists v \varepsilon \xi_A(R)$, $\exists w \varepsilon \xi_B(R)$ such that $\mathbb{B}_A(v) \cap \mathbb{B}_B(w) = \phi$. For any $A' \supseteq A$, it follows immediately that $\pi_{A'}$ refines π_A, i.e., each block in $\pi_{A'}$ is a subset of some block in π_A. Hence $\exists v' \varepsilon \xi_{A'}(R)$ such that $\mathbb{B}_{A'}(v') \subseteq \mathbb{B}_A(v)$, and $\exists w' \varepsilon \xi_{B'}(R)$ such that $\mathbb{B}_{B'}(w') \subseteq \mathbb{B}_B(w)$. As $\mathbb{B}_A(v) \cap \mathbb{B}_B(w) = \phi$, we have $\mathbb{B}_{A'}(v) \cap \mathbb{B}_{B'}(w') = \phi$ and therefore $A' \Delta_B B'$.

Theorem 2, which says that dependence is preserved by supersets, has the following "dual" statement which says that independence is preserved by subsets, that is:

<u>Theorem 3</u>: Let $A,B \subseteq D$. If $A \not\Delta_R B$ then, $\forall A' \subseteq A$, $\forall B' \subseteq B$, $\mid A' \not\Delta_R B'$.

<u>Proof</u>: Suppose to the contrary, i.e., there is a subset A' of A

and a subset B' of B such that $A' \Delta_R B'$. Then, by Theorem 2, $A \Delta_R B$. contradicting the assumption that $A \not\Delta_R B$.

The utility of Theorems 2 and 3 is that additional dependencies and independencies can be inferred from those already known. Finally, as observed earlier for finite index sets (see [3], p. 39, Theorem 3), the notion of R-independence, that is, the complement of the relation Δ_R, is closely related to the notion of a Cartesian product. More precisely,

<u>Theorem 4</u>: Let $A,B \subseteq D$ be disjoint coordinate sets, and let $\psi : \xi_{A \cup B}(R) \to \xi_A(R) \times \xi_B(R)$ be a mapping such that $\forall r \varepsilon R[\psi(\xi_{A \cup B}(r)) = (\xi_A(r), \xi_B(r))]$. (Such a map ψ always exists and is unique.) Then $A \not\Delta_R B$ if and only if $\psi(\xi_{A \cup B}(R)) = \xi_A(R) \times \xi_B(R)$.

<u>Proof</u>: Suppose $A \Delta_R B$. It suffices to show that ψ is onto. Let $v \varepsilon \xi_A(R)$, $w \varepsilon \xi_B(R)$. By the definition of R-independence, $\exists r \varepsilon R[\xi_B(r) = w$ and $\xi_A(r) = v]$. Accordingly $\psi(\xi_{A \cup B}(r)) = (\xi_A(r), \xi_B(r)) = (v,w)$. Conversely, suppose ψ is onto. Then $\forall v \varepsilon \xi_A(R)$ and $\forall w \varepsilon \xi_B(R)$, $\exists r \varepsilon R[\psi(\xi_{A \cup B}(r)) = (v,w)]$. Hence, $\exists r \varepsilon R[\xi_B(r) = w$ and $\xi_A(r) = v]$, i.e., $A \not\Delta_R B$.

An even stronger link between functional independence and Cartesian products is developed in the following subsection.

3.1.3 <u>R-dependent Coordinate Sets</u>

When examining the nature of a structured set R, it is often convenient to identify coordinate sets C ($C \subseteq D$) for which R-dependencies exist among the subsets of C. In the terminology of generalized dependence relations (see [Naylor]), such a set C is referred to as being "dependent" (in itself). When C is finite, this concept can be defined rather naturally, as was done during the previous reporting period (see [3], p. 37, Def. 5). However, when C is infinite (which is now a possibility since D may be infinite) the choice of an appropriate definition of "self-dependence" is less clear. On examining the alternatives, our choice was dictated by the desire to have a constructive test for R-dependence, even when C is infinite. Accordingly, the notion of self-dependence is formally defined as follows.

Definition 4: If R is a structured set indexed by D and $C \subseteq D$ then C is R-dependent if there exist finite sets $A, B \subseteq C$ with $A \cap B = \phi$ such that $A \Delta_R B$. C is R-independent if C is not R-dependent.

The requirement that the subsets A and B be finite provides the kind of constructive test referred to above. This is analogous to what is done in linear algebra where a dependent set of vectors must contain a finite subset for which some linear combination yields the zero vector of the space. The requirement that $A \cap B = \phi$ insures that C is not regarded as R-dependent simply because some subset of C depends on itself.

Applying Theorem 4, an R-independent set C can be characterized in terms of the algebraic structure of $\xi_C(R)$ as follows. (This characterization reduces to Theorem 4, p.41 in [3] when D is assumed to be finite.)

Theorem 5: If R is a structured set indexed by D and $C \subseteq D$, then C is R-independent if and only if $\xi_C(R) = \underset{d \varepsilon D}{\times} \xi_d(R)$.

Proof: Suppose that $\xi_C(R) = \underset{d \varepsilon C}{\times} \xi_d(R)$. Let A and B be finite disjoint subsets of C. Then $\xi_A(R) = \underset{d \varepsilon A}{\times} \xi_d(R), \xi_B(R) = \underset{d \varepsilon B}{\times} \xi_d(R)$ and

$\xi_{A\cup B}(R) = \mathop{\mathsf{X}}\limits_{d\varepsilon A\cup B} \xi_d(R)$. Let ψ be the unique coordinate mapping described in Theorem 4. Then $\psi(\xi_{A\cup B}(R)) = \psi(\mathop{\mathsf{X}}\limits_{d\varepsilon A\cup B} \xi_d(R)) = (\mathop{\mathsf{X}}\limits_{d\varepsilon A} \xi_d(R)) \times (\mathop{\mathsf{X}}\limits_{d\varepsilon B} \xi_d(R)) = \xi_A(R) \times \xi_B(R)$. Thus by Theorem 4, $A \not\Delta_R B$. Since this is true for arbitrary, finite, disjoint sets $A,B\subseteq C$, this implies that C is R-independent. Conversely, suppose C is R-independent, that is, for all finite sets $A,B\subseteq C$ such that $A\cap B = \phi$, $A \Delta_R B$. Relabel the elements $c_1, \ldots, c_m, \ldots$ of C as 1, ..., m, ... Then, in particular, $\{1\} \Delta_R \{2\}$. Applying Theorem 4 with $A = \{1\}$ and $B = \{2\}$, $\psi(\xi_{\{1,2\}}(R)) = \xi_1(R) \times \xi_2(R)$. Because 1 is the first coordinate in C, ψ is just the identity function, that is, $\psi(\xi_{\{1,2\}}(R)) = \xi_{\{1,2\}}(R) = \xi_1(R) \times \xi_2(R)$. Now take $A = \{1,2\}$ and $B = \{3\}$. Then $\{1,2\} \Delta_R \{3\}$ so $\psi(\xi_{\{1,2,3\}}(R)) = \xi_{\{1,2,3\}}(R)) = \xi_{\{1,2\}}(R) \times \xi_{\{3\}}(R)$ by Theorem 4. Once again ψ is just the identity, so $\xi_{\{1,2,3\}}(R) = \xi_{\{1,2\}}(R) \times \xi_3(R)$, and, by substitution, $\xi_{\{1,2,3\}}(R) = \xi_1(R) \times \xi_2(R) \times \xi_3(R)$. Continuing in this fashion, we can conclude that $\xi_C(R) = \mathop{\mathsf{X}}\limits_{d\varepsilon C} \xi_d(R)$.

<u>Corollary</u>: If R is indexed by D then D is R-independent if and only if R is a Cartesian product, that is, $R = \mathop{\mathsf{X}}\limits_{d\varepsilon D} \xi_d(R)$.

The "if" part of the corollary says that, whenever R is Cartesian, the index set D must be completely free of R-dependencies (in the sense of Definition 4), as one would expect given the original definition of R-dependence (Definition 3). The "only if" part is a little more surprising in that the absence of R-dependencies among finite subsets of D (even when D is infinite) guarantees that D will have the simple structure of a Cartesian product.

3.1.4 Characterization of Structure-based Capability

In addition to the above extension of the theory of R-dependence, we have explored the role that R-dependence plays in performability evaluation and, particularly, how this concept might be used to distinguish basic differences between performability modeling (as developed under the subject grant) and traditional reliability modeling. During the previous reporting period, using an example wherein success was identified with a minimum allowable average throughput(i.e., the capability function designated γ_3 in example 1, pp. 18-20 of [3]), it was argued that capability functions are more general than the "structure functions" of phased-mission reliability modeling [9]. (A somewhat more detailed version of this argument appears in [4].) During the reporting period, however, we have found that the inherent limitations of structure functions can be much more clearly and precisely characterized via the concept of R-dependence.

To establish this characterization, suppose X_S is a base-model with state space Q and one is more interested only in the reliability of the system, that is, the accomplishment set is $A = \{0,1\}$ (where 1 denotes " success" and 0 "failure"). Then, extending the notion of a "structure-based" capability function (see [3], p.17) to include "phased missions" (see [9]), a capability function γ_S is structure-based if there exists a decomposition of T into k consecutive time periods $T_1, T_2, \ldots, T_k$ and there exist functions $\varphi_1, \varphi_2, \ldots, \varphi_k$ with $\varphi_i : Q \rightarrow \{0,1\}$ such that, for all $u \in U$,

$$\gamma_S(u) = 1 \text{ if } \varphi_i(u(t)) = 1,$$

for all $i \in \{1, 2, \ldots, k\}$ and for all $t \in T$. In the context of phased mission analysis, T_i is referred to as the i^{th} phase (of the mission) and φ_i is the structure function of the i^{th} phase. Assuming further (as does phased mission analysis) that $\varphi_i(u(t)) = 1$, for all $t \varepsilon T_i$, whenever $\varphi_i(u(t_i)) = 1$ where t_i is the end of T_i, the trajectory space U can be represented by the Cartesian product $U = Q^k$. Accordingly, if $u = (q_1, q_2, \ldots, q_k)$, then $\gamma_S(u) = 1$ iff $\varphi_i(q_i) = 1$, for all $i \varepsilon \{1, 2, \ldots, k\}$. Hence $u \varepsilon \gamma_S^{-1}(1)$ iff $\xi_i(u) \varepsilon \varphi_i^{-1}(1)$ for all $i \varepsilon \{1, 2, \ldots, k\}$ and we conclude that the set $R = \gamma_S^{-1}(1)$ of "success trajectories" is also Cartesian, i.e.,

$$R = \bigtimes_{i-1}^{k} \varphi_i^{-1}(1).$$

Conversely, whenever a capability function $\gamma_S: Q^k \to \{0,1\}$ is such that $R = \gamma_S^{-1}(1)$ is Cartesian, it admits to a structure-based formulation by choosing each φ_i such that $\varphi_i(q) = 1$ iff $q \varepsilon \overline{\xi_i(R)}$. Appealing to the corollary of Theorem 5, we have proved:

Theorem 6: Let S be a phased system with trajectory space $U = Q^k$ and capability function $\gamma_S: U \to \{0,1\}$. Then γ_S is structure-based if and only if the set of all phases $D = \{1, 2, \ldots, k\}$ is R-independent, where $R = \gamma_S^{-1}(1)$.

In other words, the absence of R-dependence between phases characterizes structure-based capability functions and, accordingly, reveals the inherent limitations of structure-based reliability analysis. Performability analysis, on the other hand, can accommodate interphase dependencies, as demonstrated by the following example.

Suppose S is a multiprocessor system with three processors where the performance in question is the average throughput (Th_{av}) of the system. Suppose further that the processors are identical, so that the processing rate of the system is determined only by the number of fault-free processors. More precisely, let us suppose that the state set of the base model is $Q = \{0,1,2\}$ where the states in Q have the following interpretation:

0: all processors fault-free
1: one processor faulty
2: two or more processors faulty

Suppose further that, relative to a maximum throughput (processing rate) τ, the throughput associated with each state is as follows:

State	Throughput
0:	τ
1:	$\tau/2$
2:	0

If the utilization period is divided into phases of equal duration and we make the pessimistic assumption that the loss of a processor during a phase will affect the throughput to the same degree as the loss of a processor at the beginning of that phase, then the trajectory space of the base model is represented by the set

$$U = \{(q_1,q_2,q_3) \mid q_i \varepsilon Q\}$$

where q_i is the state of the system at the end of phase i. For the user oriented model, suppose that the accomplishment set is $A = \{a_0,a_1,a_2\}$ where a_0 corresponds to $Th_{av} \geq 5\tau/6$, a_1 corresponds to $5\tau/6 > Th_{av} \geq \tau/2$, and a_2 corresponds to $Th_{av} < \tau/2$. Then the capability function γ_S is given by:

u	$\gamma_S(u)$
(0,0,0)	a_0
(0,0,1)	a_0
(0,0,2)	a_1
(0,1,0)	a_0
(0,1,1)	a_1
(0,1,2)	a_1
⋮	⋮
(2,2,2)	a_2.

To illustrate interphase dependence, suppose we know that the accomplishment level is a_0 and let $R = \gamma^{-1}(a_0) = \{(0,0,0), (0,0,1), (0,1,0), (1,0,0)\}$. If $u \varepsilon R$ and we know that $q_2 = 1$, then we can infer that $q_1 \neq 1$. Thus, knowledge of the state of the system at the end of the second phase has increased our knowledge about the state of the system at the end of the previous phase. More formally, in terms of Def. 3, if $A = \{1\}$, $B = \{2\}$ then $v = 1$ and $w = 1$ guarantee that $\{1\} \Delta_R \{2\}$, i.e., phase 1 R-depends on phase 2.

In general, we have found that such temporal functional dependencies arise quite naturally when accomplishment levels are associated with user-visible performance. Of particular importance is the fact that such dependencies arise in the context of aircraft computer performability evaluation, as was observed during the prototype modeling and evaluation exercise conducted during the previous period (see [3], pp. 169-170). Further evidence of this fact has been revealed by the work of the current reporting period where, in evaluating the performability of the SIFT computer (see section 3.3 of this report), we have found that there is an extensive amount of interphase dependency and, indeed, more than we had originally anticipated.

3.2 Evaluation Algorithms and Programs (METAPHOR)

Concurrent with the development of performability models, concepts, measures, and measure formulations, we have proceeded with the development of evaluation algorithms and prototype evaluation tools for the purpose of investigating the feasibility of our overall approach. In particular, we are referring here to the software package called METAPHOR (Michigan Evaluation Aid for Perphormability), whose development was initiated during the previous reporting period (see [3], Section 3.5.8.1). The following sections discuss this package and the tools it contains. Detailed documentation of METAPHOR is currently in progress and will be included in the next Semi-Annual Status Report. Section 3.2.1 expands upon the objectives and abilities of METAPHOR, while Section 3.2.2 describes its use. Finally, Section 3.2.3 examines the internal structure of METAPHOR.

3.2.1 Objectives and Abilities

We envision METAPHOR as ultimately containing all the programmed tools necessary to realize a complete performability evaluation. These include aids for a) constructing the model hierarchy, b) determining the interlevel translations and their inverses, c) determining the base model trajectory sets associated with accomplishment levels , and d) evaluating the probabilities of these trajectory sets (i.e., the sets $\gamma^{-1}(a)$ for each $a \in A$). In addition, because of METAPHOR's ability to provide instruction via devices such as the HELP command, we view METAPHOR as a performability evaluation tutor.

Of the above tools, METAPHOR currently contains substantial elements of the last two, i.e., routines to calculate the probabilities of trajectory sets in addition to some tutorial capabilities. That is, once the analyst has derived the interphase and intraphase transition (P and H) matrices along with the corresponding trajectory sets of each accomplishment level, METAPHOR can then be used to calculate the probability of each accomplishment level. Presently, METAPHOR also has the ability to compute certain classes of transition matrices given such information as the structure of the components (e.g., whether computer modules are connected in a Triple Modular Redundant (TMR) fashion), and the failure rates of those components and the duration of the phase.

METAPHOR's tutorial facilities are based on an extensive repertoire of replies to HELP requests, along with preprogrammed series of questions relating to specific topics. This last feature is intended to aid a person who is learning to use the evaluation programs .

3.2.2. Use of METAPHOR

This section contains a summary of the commands and options currently implemented in METAPHOR. These are HELP, BRIEF, ECHO, EXIT, DATA, ALTER, GIVEN, DEDFAIL, NFAIL, IDENTITY, COM, and CALC. More detailed documentation of these items is currently in progress and will be reported in the next Semi-Annual Status Report.

When METAPHOR is first run, an initial heading is printed, followed by a prompt sign:

```
     METAPHOR


MICHIGAN EVALUATION AID FOR PERPHORMABILITY
              VERSION 2
                5/78


TYPE HELP FOR ASSISTANCE
⎕:
```

The quad followed by a colon is the prompt symbol indicating that METAPHOR is ready for some form of input. Three types of input may be entered. In response to most questions, numerical data is required, while a few questions need a yes/no type answer. The third type of input encompasses the command language of METAPHOR. Commands may be entered at any time, even in response to questions. (The present version does not recognize commands in answer to a yes/no question.) After the command is executed, the initial question is repeated (if appropriate).

If the user needs further assistance at some point in the program, he can enter HELP. This prints an explanation of what to do next or a brief discussion of the idea or concept currently being utilized. Also, if the user desires, a list of references concerning that idea or concept is printed. For example, if METAPHOR asks what type of interphase transition matrix is required, the user may type HELP to learn that four options are available: GIVEN, DEDFAIL, NFAIL, and IDENTITY. Further information, if requested, describes each option in detail.

Two commands are useful when supplying input from a source other than the user terminal (e.g., input from a disk file or from cards in batch mode). These are BRIEF and ECHO. BRIEF is used to suppress all output except the final results, while ECHO is employed to repeat the user supplied input. The conditions are activated by entering ECHO ON, ECHO OFF, BRIEF ON, or BRIEF OFF. The default is BRIEF OFF, ECHO OFF.

At any time, the program can be halted by entering EXIT. This causes an immediate termination of the program; it cannot be restarted.

Evaluation of trajectory set probabilities is accomplished using the EVAL command. This command initiates a sequence of queries by which METAPHOR receives a description of the trajectory sets and related items describing γ^{-1}. METAPHOR asks the following questions:

1) How many phases?
2) How many states in each phase?
3) What are the intraphase transition matrices (P) for each phase?
4) What are the interphase transition matrices (H) for each phase?
5) How many time-invariant variables?
6) What is the probability distribution of each time-invariant basic variable?
7) For each accomplishment level:
    a) How many Cartesian trajectory sets?
    b) For each Cartesian trajectory set:
        i) What is the initial state vector (I)?
        ii) What are the main diagonals of the characteristic matrices (G)?
        iii) What is the characteristic vector (F)?
        iv) What are the values of the time-invariant basic variables?

METAPHOR calculates the probability of each trajectory set "on the fly," i.e., as each Cartesian component is entered, its contribution to the overall probability is determined; the trajectory set is then discarded. (This method reduces the amount of storage necessary to perform the calculations.) The result in the form of a list (performability spectrum) is then printed. Also, if the result does not sum to 1, an error message is printed.

Currently, four types (or classes) of intraphase matrices (P) and two types of interphase matrices (H) can be entered. For the P matrices, these are GIVEN, IDENTITY, NFAIL, and DEDFAIL, while for the H matrices, either GIVEN or IDENTITY can be specified.

GIVEN allows the user to input a matrix row by row. IDENTITY automatically generates an identity matrix of the proper size.

DEDFAIL and NFAIL compute transition matrices for an special types of systems. Each assumes that the structure of the system is described in terms of "components" where the state of each component is either "operational" or "failed." Both DEDFAIL and NFAIL assume that all components are alike and fail independently with the same constant failure rate. Finally, components are assumed to fail permanently, i.e., once a component has failed, it remains failed for the duration of the phase. The difference between the two lies in how the states of the system are defined in terms of component status. DEDFAIL keeps track of each component in the system, i.e., whether a given component is operational or failed can be deduced from the state of the system. In METAPHOR, the most important use of DEDFAIL is in modeling a system wherein each component (e.g., processor) is dedicated to a different task (hence the name DEDFAIL). In such situations, the processing capability generally depends on the state of each component and hence the system state must convey the state of each component.

NFAIL, on the other hand, assumes that the components of the system are lumped into groups. NFAIL then keeps track only of the number of components which are operational within each of these groups. For instance, if two tasks and four processors are configured such that two processors are executing each task, then failure of either processor assigned to a given task will have the same effect on system performance. Accordingly, processors sharing the same task can be lumped, resulting in 2 groups with 2 processors per group. NFAIL is equivalent to DEDFAIL when NFAIL has n groups of 1 element each.

If at any time the user wishes to know what value METAPHOR has assigned to some variable, or if the user wishes to change the value of some variable, then the commands DATA or ALTER may be employed. DATA causes two lines of abbreviations to be printed as below:

```
⎕:
      DATA
PUT AN X BELOW EACH ITEM TO BE DISPLAYED.  HELP AVAILABLE.
NUM.PHASES  NUM.STATES  P  H  NUM.CONST.BAS.VARS  PROB.CONST.BAS.VARS
                  X                  X
NUM.ACC.LEVELS  NUM.TRAJ.SETS  I  G  F  V  PERF
           X                   X
```

The user places an X below the items he wishes to display. Each item is printed so long as it has been defined, otherwise a warning is given stating that the item has not been defined. The above abbreviations should be straightforward; note that time-invariant basic variables are referred to as "constant" basic variables (CONST.BAS.VARS), "NUM" stands for "number of," while "ACC" for "accomplishment," and "TRAJ" for "trajectory." V is the vector characterizing the time-invariant basic variables.

ALTER operates in a manner similar to DATA. One line of abbreviations is presented:

```
⎕:
      ALTER
PUT AN X BELOW EACH ITEM TO BE CHANGED.  HELP AVAILABLE.
P  H  CONST.BAS.VARS  ALL.ACC.LEVELS  PRESENT.ACC.LEVEL  I  G  F V NUM.TRAJ.SETS
         X                                    X
```

Again the user places an X below each item to be changed. This command is particularly useful if an error is made while entering data.

Two other commands are available. COM allows the user to enter lines of text as comments. METAPHOR will prompt each line with "***", after which any characters may be typed. Giving a carriage return with no characters (a null line) ends the comment section. CALC allows the user to utilize the APL calculator mode. Each line will be prompted by a question mark, a quad and then a colon as follows:

```
⎕:
      CALC
?
⎕:
```

The user is advised not to employ assignment statements (such as A ← 6), since the names of variables chosen may interfere with names of variables internal to METAPHOR. When in CALC mode, typing EXIT returns the user to his previous status, i.e., the state of the program before CALC mode was entered.

Figures 1-2 give sample METAPHOR sessions.

```
MICHIGAN EVALUATION AID FOR PERPHORMABILITY
              VERSION 2
                5/78


TYPE HELP FOR ASSISTANCE
ECHO ECHO  ON
⎕:  EVAL

NUMBER OF PHASES?
⎕:  2

NUMBER OF STATES PER PHASE?  (SPACE BETWEEN EACH NUMBER)
⎕:  3  2


SPECIFY THE P MATRICES FOR EACH PHASE, 1 PHASE AT A TIME

PHASE 1:

WHAT TYPE OF P MATRIX?
⎕:  NFAIL
ENTER PHASE LENGTH
⎕:  10
ENTER COMPONENT FAILURE RATE
⎕:  0.0001
ENTER NUMBER OF GROUPS
⎕:  1
ENTER NUMBER OF COMPONENTS PER GROUP  (SPACE BETWEEN EACH NUMBER):
⎕:  2

PHASE 2:

WHAT TYPE OF P MATRIX?
⎕:  DEDFAIL
ENTER PHASE LENGTH
⎕:  5
ENTER COMPONENT FAILURE RATE
⎕:  0.0001


SPECIFY THE H MATRICES FOR EACH PHASE, 1 PHASE AT A TIME

PHASE 1-2:

WHAT TYPE OF H MATRIX?
⎕:  GIVEN
ENTER THE MATRIX, 1 ROW AT A TIME
```

FIGURE 1

Sample METAPHOR Session

```
ROW 1:
⎕:  1  0
ROW 2:
⎕:  1  0
ROW 3:
⎕:  0  1

NUMBER OF CONSTANT BASIC VARIABLES?
⎕:  1

PROBABILITIES OF EACH CONSTANT VARIABLE?  (SPACE BETWEEN EACH NUMBER)
⎕:  0.001

NUMBER OF ACCOMPLISHMENT LEVELS?
⎕:  3

ACCOMPLISHMENT LEVEL 0
NUMBER OF TRAJECTORY SETS FOR THIS ACCOMPLISHMENT LEVEL?
⎕:  1
TRAJECTORY SET 1
ENTER THE I VECTOR (SPACE BETWEEN EACH ENTRY):
⎕:  1  0  0
PHASE 1:
ENTER THE G DIAGONAL (SPACE BETWEEN EACH ENTRY):
⎕:  1  0  0
ENTER THE F VECTOR (SPACE BETWEEN EACH ENTRY):
⎕:  1  0
ENTER THE 1 ELEMENT CONSTANT BASIC VARIABLE VECTOR  (SPACE BETWEEN EACH ENTRY):
⎕:  0

ACCOMPLISHMENT LEVEL 1
NUMBER OF TRAJECTORY SETS FOR THIS ACCOMPLISHMENT LEVEL?
⎕:  2
TRAJECTORY SET 1
ENTER THE I VECTOR (SPACE BETWEEN EACH ENTRY):
⎕:  1  0  0
PHASE 1:
ENTER THE G DIAGONAL (SPACE BETWEEN EACH ENTRY):
⎕:  0  1  0
ENTER THE F VECTOR (SPACE BETWEEN EACH ENTRY):
⎕:  1  0
ENTER THE 1 ELEMENT CONSTANT BASIC VARIABLE VECTOR  (SPACE BETWEEN EACH ENTRY):
⎕:  2
TRAJECTORY SET 2
ENTER THE I VECTOR (SPACE BETWEEN EACH ENTRY):
⎕:  1  0  0
PHASE 1:
ENTER THE G DIAGONAL (SPACE BETWEEN EACH ENTRY):
⎕:  1  0  0
ENTER THE F VECTOR (SPACE BETWEEN EACH ENTRY):
⎕:  1  0
ENTER THE 1 ELEMENT CONSTANT BASIC VARIABLE VECTOR  (SPACE BETWEEN EACH ENTRY):
⎕:  1
```

FIGURE 1 (Continued)

le METAPHOR Session

```
ACCOMPLISHMENT LEVEL 2
NUMBER OF TRAJECTORY SETS FOR THIS ACCOMPLISHMENT LEVEL?
⎕:  1
TRAJECTORY SET 1
ENTER THE I VECTOR (SPACE BETWEEN EACH ENTRY):
⎕:  1  0  0
PHASE 1:
ENTER THE G DIAGONAL (SPACE BETWEEN EACH ENTRY):
⎕:  1  1  1
ENTER THE F VECTOR (SPACE BETWEEN EACH ENTRY):
⎕:  0  1
ENTER THE 1 ELEMENT CONSTANT BASIC VARIABLE VECTOR  (SPACE BETWEEN EACH ENTRY):
⎕:  2


PERFORMABILITY FOR THIS MISSION ← 0.0009975031224  0.9985016234  0.000500873522
```

FIGURE 1 (Continued)
ple METAPHOR Session

```
        METAPHOR


MICHIGAN EVALUATION AID FOR PERPHORMABILITY
             VERSION 2
               5/78


TYPE HELP FOR ASSISTANCE
⎕:
        COM
***
*** FIGURE 2 DEMONSTRATES SOME OF THE UTILITY
*** FUNCTIONS AVAILABLE IN METAPHOR
***
⎕:
        HELP
METAPHOR IS AN INTERACTIVE SOFTWARE PACKAGE AIDING THE MODELING
AND ANALYSIS OF PERFORMABILITY.  AT PRESENT, METAPHOR IS CAPABLE
ONLY OF EVALUATING CERTAIN PERFORMABILITY MODELS.
THE COMMANDS PRESENTLY AVAILABLE ARE:  EVAL,HELP, DATA, ALTER, CALC,
          COM, BRIEF [ON|OFF], ECHO [ON|OFF], AND EXIT.
DO YOU WANT MORE HELP?
⎕:
NO
⎕:
        EVAL

NUMBER OF PHASES?
⎕:
        2

NUMBER OF STATES PER PHASE?  (SPACE BETWEEN EACH NUMBER)
⎕:
        1 2


SPECIFY THE P MATRICES FOR EACH PHASE, 1 PHASE AT A TIME

PHASE 1:

WHAT TYPE OF P MATRIX?
⎕:
        HELP
TYPE ONE OF:   GIVEN, DEDFAIL, NFAIL, IDENTITY
DO YOU WANT MORE HELP?
⎕:
NO
WHAT TYPE OF P MATRIX?
⎕:
        NFAIL
```

FIGURE 2

mple METAPHOR Session

```
ENTER PHASE LENGTH
⎕:
      2
ENTER COMPONENT FAILURE RATE
⎕:
      1
1 IS LARGE FOR A FAILURE RATE.  DO YOU WANT THIS VALUE?
⎕:
NOPE
ENTER COMPONENT FAILURE RATE
⎕:
      .001
ENTER NUMBER OF GROUPS
⎕:
      DATA
PUT AN X BELOW EACH ITEM TO BE DISPLAYED.  HELP AVAILABLE.
NUM.PHASES  NUM.STATES  P  H  NUM.CONST.BAS.VARS  PROB.CONST.BAS.VARS
      X          X      X                               X
NUM.ACC.LEVELS  NUM.TRAJ.SETS  I  G  F  V  PERF
    X                                         X

NUMBER OF PHASES IS 2
NUMBER OF STATES PER PHASE IS 1  2
P MATRICES HAVE NOT BEEN DEFINED
THE CONSTANT BASIC VARIABLES HAVE NOT BEEN DEFINED
THE NUMBER OF ACCOMPLISHMENT LEVELS NOT DEFINED
PERFORMABILITY NOT DEFINED
ENTER NUMBER OF GROUPS
⎕:
      ALTER
PUT AN X BELOW EACH ITEM TO BE CHANGED.  HELP AVAILABLE.
P   H  CONST.BAS.VARS  ALL.ACC.LEVELS  PRESENT.ACC.LEVEL  I  G  F V NUM.TRAJ.SETS
X                             X
P MATRICES ARE NOT DEFINED AT THIS TIME.
THE ACCOMPLISHMENT LEVELS ARE NOT DEFINED AT THIS TIME.
THE NUMBER OF TRAJECTORY SETS IS NOT DEFINED AT THIS TIME.

⎕:
      EXIT
```

FIGURE 2 (Continued)

Sample Metaphor Session

3.2.3 Internal Structure

This section presents a brief overview of the way METAPHOR operates internally. Currently, METAPHOR is in its second major version. It is written in APL and contains approximately 2000 lines of code. The package is highly modular, with about 60 APL functions (somewhat analogous to FORTRAN subroutines). Figure 3 lists the currently available functions.

Although APL does not lend itself readily towards structured programming, a substantial effort was made to make the package easily readable and maintainable. Thus, for example, specific conventions regarding the names of functions, variables, and labels have been established.

Various methods of control and information exchange among the various functions are employed. For instance, there is a versatile input function which determines whether the item (or items) entered by the user is a command or data. If a command is given, the proper corresponding functions are then called. If data is given, a check is made to insure that it is of the correct size. Other functions check to see if the data is consistent, e.g., if a probability distribution is to be input, then it must sum to one. Some of the user assistance commands, namely HELP, ALTER, and DATA, have somewhat involved control mechanisms. For example, METAPHOR must be aware of which function it is executing in order to correctly respond to HELP requests.

More complete documentation of METAPHOR's internal structure is currently underway and will be included in the next Semi-Annual Status Report.

Main Functions

METAPHOR
DECLAREMETAPHOR
METINFO

Command Functions

COMMANDALTER
COMMANDBRIEF
COMMANDCALC
COMMANDCOM
COMMANDDATA
COMMANDECHO
COMMANDEVAL
COMMANDHELP

Command Support Functions

BRIEF
ECHO
GETALTERVECTOR
GAVINFO
GETDATAVECTOR
GDVINFO

Command EVAL Implementation Functions

GETNUMPHASES
GNPINFO
GETSTATES
GSINFO
GETPMATRICES
GETHMATRICES
GETNUMBASICVARIABLES
GNBVINFO
GETBASICVARIABLES
GBVINFO
GETNUMACCLEV
GNAINFO
GETPERF
PUTPERFORMABILITY

Matrix Generator Functions

GENERATEHMATRIX
GHMINFO
GENERATEPMATRIX
GPMINFO
GDEDFAIL
GDINFO
GGIVEN
GGINFO
GIDENTITY
GNFAIL
GNINFO

Trajectory Set Evaluation Functions

GETACCLEVPROB
GETNUMTRAJSETS
GNTSINFO
GETIVECTOR
GIVINFO
GETGMATRICES
GGMINFO
GETFVECTOR
GFVINFO
GETVVALUES
GVVINFO
CALCTRAJPROB

I/O and Checking Functions

INPUT
INYES
CHECKBIN
CHECKPOSI
CHECKPROB
CHECKTRI
PRINT
PRINTQUAD

APL Support Function

ENCODE

FIGURE 3

Currently available METAPHOR functions

3.3 Performability Evaluation of the SIFT Computer in an Air Transport Mission

During the reporting period, we have completed a relatively comprehensive performability modeling and evaluation exercise involving the SIFT computer [8] as it might operate in the environment of a transoceanic air transport mission. In carrying out this exercise, we have attempted to strike a balance between simplicity and reality that permits all aspects of the methodology to be demonstrated in the context of a meaningful evaluation problem. In particular, reality was stressed in the construction of higher level models, where our assumptions are based on the study of computational requirements made by R. S. Ratner, et al. [7]. Simplicity was stressed in our choice of a bottom level Markov model of the SIFT computer (similar to "Model I" used by SRI; see [8], p. 151) in order to reduce the complexity of the performability calculations. However, more realistic bottom models (e.g., SRI "Model IV") are compatible with the remainder of the hierarchy and could replace the simpler bottom model.

The description of this effort is organized as follows. First, the performance model (accomplishment set), two upper level models (mission level and aircraft functional task level), as well as the interconnections between them, are presented in Section 3.3.1. Section 3.3.2 then introduces the bottom model of the SIFT computer and describes the interlevel translation between it and the functional task level. Both the algorithm by which tasks are allocated to the computer as well as a Markov model describing the hardware are discussed. Derivations of the base model trajectory sets (associated with each level of accomplishment) are described in Section 3.3.3, and derivation of computer model transition matrices is described in Section 3.3.4. Finally, the numerical results of the evaluation exercise are presented in Section 3.3.5.

3.3.1 Higher Level Models

3.3.1.1 Performance Model

The total system S = (C, E) considered is the SIFT computer C operated in the environment E of a transoceanic flight of a commercial aircraft. The mission of the total system can be characterized as follows:

> "Transport passengers between two points (separated by an ocean) with safety, with no significant change of mission, with no significant operational penalties or stress on crew or Air Traffic Control, and with no significant economic penalties."

Examining this statement in more detail, total system performance can be described in terms of four attributes: safety, no change in mission profile, no operational penalties, and no economic penalties. Attributes similar to these have been used by Ratner, et al. [7] to distinguish the "criticalities" of various aircraft functions. To determine the accomplishment set A for the performance variable Y_S, we assume that safety is the most important attribute, i.e., safe flights have the greatest worth, the remaining attributes being worth successively less in the order they are listed above. These assumptions agree with the "reliability requirements" (see [7], p.7) associated with corresponding criticality levels. We assume further that safety is worth considerably more than no change in mission profile, which in turn is worth considerably more than no operational penalties, etc. Thus, for example, if there is a change in mission profile (i.e., loss of the attribute "no change in mission profile") then the presence or absence of lower worth attributes (i.e., "no operational penalties" and "no economic penalties") will have a negligible effect on the worth of the mission outcome.

With the above assumptions, the following set suffices to describe the relevant levels of accomplishment:

$$A = \{a_0, a_1, a_2, a_3, a_4\}$$

where each level has the following general definition:

a_0 = no economic penalties, no operational penalties, no change in mission profile, and no fatalities,

a_1 = economic penalties, no operational penalties, no change in mission profile, and no fatalities,

a_2 = operational penalties, no change in mission profile, and no fatalities,

a_3 = change in mission profile, and no fatalities,

a_4 = fatalities.

Thus, by accounting for the relative importance of various attributes, the number of relevant levels of accomplishment is reduced from 16 (the number of subsets of the set of 4 attributes) to 5. On the other hand, the information regarding relative worths is not essential, i.e., the evaluation could be carried out relative to a 16 level accomplishment set.

For this accomplishment set, we then developed a hierarchical model of the total system comprised of three levels:

Level 0: Mission Level

Level 1: Aircraft Functional Task Level

Level 2: Computer Level.

Construction of this model proceeded in a top-down manner (i.e., level 0 → level 1 → level 2) as generally discussed in earlier reports (see [3], pp. 81-82, for example). The subsections that follow describe this hierarchy, i.e., the models at each level and the interlevel translations between adjacent models.

3.3.1.2 Mission Level Model

The mission level model (level 0) describes the total system performance in terms closely related to the accomplishment set A.

Formally, this model is a single variable random process $Z = X_h^0$ taking values in the state space

$$Q^0 = \{0,1\}^4$$

where a state $q = (q_1, q_2, q_3, q_4) \in Q^0$ is interpreted as follows:

$$q_1 = \begin{cases} 0 & \text{if the mission has no economic penalties} \\ 1 & \text{otherwise,} \end{cases}$$

$$q_2 = \begin{cases} 0 & \text{if the mission has no operational penalties} \\ 1 & \text{otherwise,} \end{cases}$$

$$q_3 = \begin{cases} 0 & \text{if the mission has no change in mission profile} \\ 1 & \text{otherwise,} \end{cases}$$

$$q_4 = \begin{cases} 0 & \text{if the mission is safe} \\ 1 & \text{otherwise.} \end{cases}$$

If ξ_i is the projection of Q^0 onto its i^{th} coordinate, we let z_i denote the random variable $\xi_i Z$, i.e.,

$$Z = (z_1, z_2, z_3, z_4).$$

As a mnemonic aid, these variables are alternatively referred to as follows:

z_1 = ECONOMICS
z_2 = OPERATIONS
z_3 = PROFILE
z_4 = SAFETY.

Because the level 0 model consists of a single random variable, the trajectory space U^0 (see [3], p. 20) coincides with the state space, i.e.,

$$U^0 = Q^0 = \{0,1\}^4.$$

Table 1 specifies the inverse of the interlevel translation κ_0 or, what is the same, the partial capability function γ_0 (see [3], p. 26). Because of the inability of computer output to denote subscripts, the accomplishment level indices are placed in parenthesis after the letter "a." For example, a_3 is written a(3). This is similar to the method used in FORTRAN and other computer languages to specify array subscripts and should cause no confusion. The "*" notation of Table 1 represents a "don't care" situation and signifies that any valid entry is acceptable.

Table 1
Page 1 of 1

LEVEL 0 TO PERFORMANCE LEVEL INTERLEVEL TRANSLATION

(LEVEL 0
INVERSE PARTIAL CAPABILITY FUNCTION)

Performance Level	Level 0 Trajectory Sets
a (0)	$\begin{bmatrix}0\\0\\0\\0\end{bmatrix}\begin{matrix}\text{ECONOMICS}\\\text{OPERATIONS}\\\text{PROFILE}\\\text{SAFETY}\end{matrix}$
a (1)	$\begin{bmatrix}1\\0\\0\\0\end{bmatrix}\begin{matrix}\text{ECONOMICS}\\\text{OPERATIONS}\\\text{PROFILE}\\\text{SAFETY}\end{matrix}$
a (2)	$\begin{bmatrix}*\\1\\0\\0\end{bmatrix}\begin{matrix}\text{ECONOMICS}\\\text{OPERATIONS}\\\text{PROFILE}\\\text{SAFETY}\end{matrix}$
a (3)	$\begin{bmatrix}*\\*\\1\\0\end{bmatrix}\begin{matrix}\text{ECONOMICS}\\\text{OPERATIONS}\\\text{PROFILE}\\\text{SAFETY}\end{matrix}$
a (4)	$\begin{bmatrix}*\\*\\*\\1\end{bmatrix}\begin{matrix}\text{ECONOMICS}\\\text{OPERATIONS}\\\text{PROFILE}\\\text{SAFETY}\end{matrix}$

3.3.1.3 Aircraft Functional Task Level Model

To determine an appropriate model at the aircraft functional task level, we assume the following characteristics regarding the aircraft to be used in the mission. (See [7] for more details regarding these specific aircraft functions.)

a) The aircraft has an Aircraft Integrated Data System (AIDS) which continuously executes in-flight analyses of various on-board data. This information is economically useful to the airline for assessing aircraft performance and for scheduling maintenance. Thus, loss of AIDS results in an economic setback to the air carrier.

b) The aircraft has two means of navigation. The first involves an inertial guidance system (INERTIAL) which will operate at any point regardless of latitude, while the second means involves an air data system (AIR DATA) along with two radio beacon systems: Very-High Frequency Omniranges (VOR) and Distance Measuring Equipment (DME). We assume that the signals generated by the VOR/DME systems will not be receivable by aircraft more than 250 nautical miles from a transmitting station, and in particular, more than 250 nautical miles from land. The AIR DATA function is required to support the VOR/DME function.

c) If the aircraft loses its inertial system before entering a region where it cannot receive VOR/DME signals, (especially an oceanic region on a transoceanic mission), the plane will return to its origin. We make the simplifying assumption that if the plane must make such a diversion, the plane returns safely to its origin with no further incidents. This assumption is made because the theory to support the use of multiple, state-dependent utilization periods has not yet been developed. Such a diversion is considered a change in mission profile.

d) If the aircraft loses its inertial system while out of range of VOR/DME, then the plane loses all navigational capability. Likewise, if the aircraft loses its INERTIAL system and its capability to analyze VOR/DME-AIR DATA information (i.e., either the VOR/DME function or the AIR DATA function), then the aircraft loses all navigational capability. Such a loss of navigation will be considered a change in mission profile.

e) If the aircraft loses either its AIR DATA function or its VOR/DME function, then the loss is considered an operational penalty. Of course, if both functions fail, a change in mission profile may occur. (See d) above.)

f) The aircraft has an autoland system (AUTOLAND) which, if working, will land the plane in any weather. This system is used only in Category III weather. The AUTOLAND system requires the results of INERTIAL computations as well as AUTOLAND computations. If at the initiation of landing, the destination airport has Category III weather and the aircraft does not possess the AUTOLAND capability, then a diversion is made to another airport. Such a diversion is considered a change in mission profile.

g) If at the initiation of landing, the destination airport has Category III weather, and the aircraft has the AUTOLAND capability, then loss of AUTOLAND during landing will cause the plane to crash, resulting in an unsafe mission.

h) The aircraft has active flutter control (ACTIVE FLUTTER CONTROL), attitude control (ATTITUDE CONTROL), and engine control (ENGINE CONTROL) functions, all of which are critical to the airworthiness of the plane. Loss of any one of these functions entails fatalities and, hence, an unsafe mission.

i) The onboard computer is involved actively in all aircraft functions mentioned above. Furthermore, the computer is involved in no other tasks.

Under the above assumptions, we have the following (worst case) conditions relating functional tasks to the mission variables z_1, z_2, z_3, z_4 discussed in the previous section:

$$z_1 = \text{ECONOMICS} = \begin{cases} 0 & \text{if AIDS works for the entire mission} \\ 1 & \text{if AIDS fails at some point in the mission,} \end{cases}$$

$$z_2 = \text{OPERATIONS} = \begin{cases} 0 & \text{if VOR/DME and AIR DATA both work for the entire mission} \\ 1 & \text{if VOR/DME or AIR DATA fail at some point in the mission,} \end{cases}$$

z_3 = PROFILE =

0 if i) INERTIAL works through the initiation of landing and AUTOLAND works at the initiation of landing, or ii) INERTIAL works until the plane is near enough to its destination to receive VOR/DME and the weather at the initiation of landing is not Category III, or iii) INERTIAL works through the initiation of landing and the weather at the initiation of landing is not Category III,

1 if i) INERTIAL and either VOR/DME or AIR DATA fail when the plane is near its destination, or ii) INERTIAL fails when the plane is near its source, or iii) INERTIAL fails when the plane is out of range of VOR/DME signals, or iv) AUTOLAND fails at the initiation of landing and the weather at the initiation of landing is Category III,

z_4 = SAFETY =

0 if either i) AUTOMATIC FLUTTER CONTROL, ENGINE CONTROL, and ATTITUDE CONTROL work during the entire mission; INERTIAL works while the aircraft is close to its source (until the plane is out of range of VOR/DME); and at the initiation of landing and one of the following is true:
a) the weather is not Category III, or
b) the weather is Category III but either AUTOLAND or INERTIAL does not work, or
c) the weather is Category III, and both AUTOLAND and INERTIAL work through the conclusion of the landing,
or ii) AUTOMATIC FLUTTER CONTROL, ENGINE CONTROL, and ATTITUDE CONTROL work while the aircraft is close to its source (until the plane is out of range of VOR/DME), but INERTIAL does not work at some point during the same interval,

1 if either i) AUTOMATIC FLUTTER CONTROL, ENGINE CONTROL, or ATTITUDE CONTROL do not work at some point during the mission, or ii) at the initiation of landing , the weather is Category III, and AUTOLAND and INERTIAL work, but then during landing, either AUTOLAND or INERTIAL fail.

Hence, the model at the aircraft functional task level involves the following eight aircraft tasks along with a single environment variable:

AIDS
VOR/DME
AIR DATA
INERTIAL
AUTOLAND
ACTIVE FLUTTER CONTROL
ENGINE CONTROL
ATTITUDE CONTROL
WEATHER (environment).

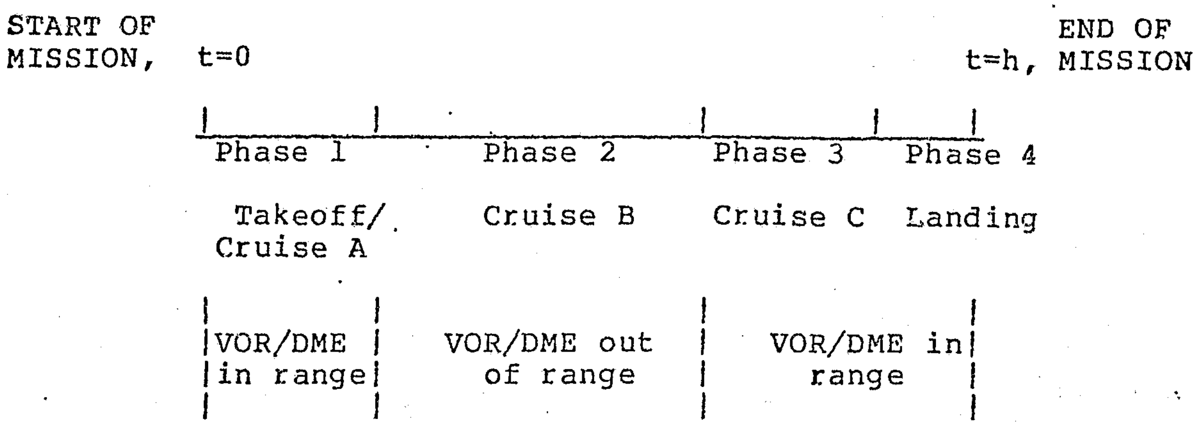
Also, because of the considerations regarding the range of the VOR/DME and the initiation of landing, four phases are appropriate:

Phase 1 = Takeoff/cruise until VOR/DME out of range,
Phase 2 = Cruise until VOR/DME in range again,
Phase 3 = Cruise until landing is to be initiated, and
Phase 4 = Landing,

where their descriptions are abbreviated as follows:

Phase 1 = Takeoff/Cruise A
Phase 2 = Cruise B
Phase 3 = Cruise C
Phase 4 = Landing.

Graphically, the utilization period is decomposed as follows:



Formally, the aircraft level model is a random process Y with four random variables:

$$Y = \{X_a^1, X_b^1, X_c^1, X_h^1\}$$

where a is the time at which Cruise A ends, b is the time at which Cruise B ends, c is the time at which Cruise C ends, and h is the time at which the landing ends (since the utilization period is [0,h]). The state space for each phase is

$$Q^1 = \{0,1\}^9$$

where a state $q = (q_1, q_2, q_4, q_5, q_6, q_7, q_8, q_9)$ in Q^1 is interpreted as follows:

$$q_1 = \begin{cases} 0 & \text{if AIDS works during the entire phase} \\ 1 & \text{if AIDS fails at some point during the phase,} \end{cases}$$

$$q_2 = \begin{cases} 0 & \text{if VOR/DME works during the entire phase} \\ 1 & \text{if VOR/DME fails at some point during the phase,} \end{cases}$$

$$q_3 = \begin{cases} 0 & \text{if AIR DATA works during the entire phase} \\ 1 & \text{if AIR DATA fails at some point during the phase,} \end{cases}$$

$$q_4 = \begin{cases} 0 & \text{if INERTIAL works during the entire phase} \\ 1 & \text{if INERTIAL fails at some point during the phase,} \end{cases}$$

$$q_5 = \begin{cases} 0 & \text{if AUTOLAND works during the entire phase} \\ 1 & \text{if AUTOLAND fails at some point during the phase,} \end{cases}$$

$$q_6 = \begin{cases} 0 & \text{if ACTIVE FLUTTER CONTROL works during the entire phase} \\ 1 & \text{if ACTIVE FLUTTER CONTROL fails at some point during the phase,} \end{cases}$$

$$q_7 = \begin{cases} 0 & \text{if ENGINE CONTROL works during the entire phase} \\ 1 & \text{if ENGINE CONTROL fails at some point during the phase,} \end{cases}$$

$$q_8 = \begin{cases} 0 & \text{if ATTITUDE CONTROL works during the entire phase} \\ 1 & \text{if ATTITUDE CONTROL fails at some point during the phase,} \end{cases}$$

$$q_9 = \begin{cases} 0 & \text{if non-Category III weather at the initiation of landing} \\ 1 & \text{otherwise.} \end{cases}$$

Using the array representation discussed in the Third Semi-Annual Status Report [3], the process Y is written as a matrix of random variables

$$Y = \left[\begin{array}{cccc} y_{11} & y_{12} & y_{13} & y_{14} \\ y_{21} & y_{22} & y_{23} & y_{24} \\ y_{31} & y_{32} & y_{33} & y_{34} \\ y_{41} & y_{42} & y_{43} & y_{44} \\ y_{51} & y_{52} & y_{53} & y_{54} \\ y_{61} & y_{62} & y_{63} & y_{64} \\ y_{71} & y_{72} & y_{73} & y_{74} \\ y_{81} & y_{82} & y_{83} & y_{84} \\ y_{91} & y_{92} & y_{93} & y_{94} \end{array}\right] \begin{array}{l} \text{AIDS} \\ \text{AIR DATA} \\ \text{VOR/DME} \\ \text{INERTIAL} \\ \text{AUTOLAND} \\ \text{ACTIVE FLUTTER CONTROL} \\ \text{ENGINE CONTROL} \\ \text{ATTITUDE CONTROL} \\ \text{WEATHER.} \end{array}$$

Here y_{ij} is the i^{th} coordinate of the j^{th} variable in Y (e.g., y_{23} is the AIR DATA coordinate of X_c^1). In the discussion of Section 3.3.3, an alternate notation for y_{ij} will sometimes be employed: if I is the name of row i (as indicated given above), then y_{ij} will be written I(j), e.g., y_{23} = AIR DATA (3). Accordingly, the trajectory space for the level 1 model is

$$\begin{aligned} U^1 &= \{0,1\}^9 \times \{0,1\}^9 \times \{0,1\}^9 \times \{0,1\}^9 \\ &= \{0,1\}^{36} \end{aligned}$$

whose elements are represented as 9×4 matrices over {0,1}.

Using the above information, the translation between the mission level (level 0) and the aircraft functional task level (level 1) was formulated, i.e., the inverse κ_1^{-1} of the level 1 to level 0 interlevel translation $\kappa_1 : U^1 \to U^0$ (see [3], p.24). Employing the method described in the previous report ([4],pp. 96-103) $\kappa_1^{-1}(z)$ for some mission outcome z can then be expressed as the intersection of the component inverses $(\xi_i\kappa_1)^{-1}$, i.e.,

$$\kappa_1^{-1}(z) = (\xi_1\kappa_1)^{-1}(z_1) \cap (\xi_2\kappa_1)^{-1}(z_2) \cap (\xi_3\kappa_1)^{-1}(z_3) \cap (\xi_4\kappa_1)^{-1}(z_4)$$

$$\begin{aligned} &= (\xi_{\mathrm{ECONOMICS}}\kappa_1)^{-1}(\mathrm{ECONOMICS}) \\ &\quad \cap(\xi_{\mathrm{OPERATIONS}}\kappa_1)^{-1}(\mathrm{OPERATIONS}) \\ &\quad \cap(\xi_{\mathrm{PROFILE}}\kappa_1)^{-1}(\mathrm{PROFILE}) \\ &\quad \cap(\xi_{\mathrm{SAFETY}}\kappa_1)^{-1}(\mathrm{SAFETY}) \end{aligned}$$

where

$$z = \begin{array}{ll} \left[\begin{array}{c} z_1 \\ z_2 \\ z_3 \\ z_4 \end{array}\right] & \begin{array}{l} \mathrm{ECONOMICS} \\ \mathrm{OPERATIONS} \\ \mathrm{PROFILE} \\ \mathrm{SAFETY} \end{array} \end{array}$$

is some mission level trajectory.

Table 2 shows the component inverses $(\xi_i\kappa_i)^{-1}$ (i=1,2,3,4) of the interlevel translation κ_1. The first column of the table names the coordinate being considered, that is, one of ECONOMICS (z_1), OPERATIONS (z_2), PROFILE (z_3), or SAFETY (z_4), while the second column gives the value of that coordinate (either 0 or 1). The third column presents a level 1 trajectory set that maps into the given level 0 coordinate value. For coordinate i and value v the union of all the indicated Cartesian trajectory sets is the set $(\xi_i\kappa_1)^{-1}(v)$. Thus, for example, the trajectory set which corresponds to SAFETY=0, i.e., the set $(\xi_3\kappa_1)^{-1}(0)$, is

COORDINATE INVERSES OF
LEVEL 2 TO LEVEL 1 TRANSLATION

Table 2
Page 1 of 10

LEVEL 0		LEVEL 1					
COORDINATE	VALUE	TRAJECTORY SETS					NAME
ECONOMICS	0	0	0	0	0	AIDS	A
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		*	*	*	*	INERTIAL	
		¢	¢	*	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	
ECONOMICS	1	1	*	*	*	AIDS	b
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		*	*	*	*	INERTIAL	
		¢	¢	*	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	
ECONOMICS	1	0	1	*	*	AIDS	c
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		*	*	*	*	INERTIAL	
		¢	¢	*	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	
ECONOMICS	1	0	0	1	*	AIDS	d
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		*	*	*	*	INERTIAL	
		¢	¢	*	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

COORDINATE INVERSES OF LEVEL 2 TO LEVEL 1 TRANSLATION

LEVEL 0		LEVEL 1					NAME
COORDINATE	VALUE	TRAJECTORY SETS					
ECONOMICS	1	0	0	0	1	AIDS	e
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		*	*	*	*	INERTIAL	
		∅	∅	*	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		∅	∅	*	∅	WEATHER	
OPERATIONS	0	*	*	*	*	AIDS	A
		0	0	0	0	VOR/DME	
		0	0	0	0	AIR DATA	
		*	*	*	*	INERTIAL	
		∅	∅	*	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		∅	∅	*	∅	WEATHER	
OPERATIONS	1	*	*	*	*	AIDS	b
		1	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		*	*	*	*	INERTIAL	
		∅	∅	*	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		∅	∅	*	∅	WEATHER	
OPERATIONS	1	*	*	*	*	AIDS	c
		0	1	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		*	*	*	*	INERTIAL	
		∅	∅	*	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		∅	∅	*	∅	WEATHER	

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

COORDINATE INVERSES OF LEVEL 2 TO LEVEL 1 TRANSLATION

Table 2
Page 3 of 10

ORIGINAL PAGE OF POOR QUALITY

LEVEL 0 COORDINATE	VALUE	LEVEL 1 TRAJECTORY SETS					NAME
OPERATIONS	1	*	*	*	*	AIDS	d
		0	0	1	*	VOR/DME	
		*	*	*	*	AIR DATA	
		*	*	*	*	INERTIAL	
		¢	¢	*	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	
OPERATIONS	1	*	*	*	*	AIDS	e
		0	0	0	1	VOR/DME	
		*	*	*	*	AIR DATA	
		*	*	*	*	INERTIAL	
		¢	¢	*	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	
OPERATIONS	1	*	*	*	*	AIDS	f
		0	0	0	0	VOR/DME	
		1	*	*	*	AIR DATA	
		*	*	*	*	INERTIAL	
		¢	¢	*	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	
OPERATIONS	1	*	*	*	*	AIDS	g
		0	0	0	0	VOR/DME	
		0	1	*	*	AIR DATA	
		*	*	*	*	INERTIAL	
		¢	¢	*	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	
OPERATIONS	1	*	*	*	*	AIDS	h
		0	0	0	0	VOR/DME	
		0	0	1	*	AIR DATA	
		*	*	*	*	INERTIAL	
		¢	¢	*	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

COORDINATE INVERSES OF LEVEL 2 TO LEVEL 1 TRANSLATION

Table 2
Page 4 of 10

LEVEL 0		LEVEL 1 TRAJECTORY SETS					NAME
COORDINATE	VALUE						
OPERATIONS	1	*	*	*	*	AIDS	i
		0	0	0	0	VOR/DME	
		0	0	0	1	AIR DATA	
		*	*	*	*	INERTIAL	
		¢	¢	*	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	
MISSION PROFILE	0	*	*	*	*	AIDS	A
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		0	0	0	*	INERTIAL	
		¢	¢	0	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	
MISSION PROFILE	0	*	*	*	*	AIDS	B
		*	*	0	*	VOR/DME	
		*	*	0	*	AIR DATA	
		0	0	1	*	INERTIAL	
		¢	¢	*	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		¢	¢	0	¢	WEATHER	
MISSION PROFILE	0	*	*	*	*	AIDS	C
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		0	0	0	*	INERTIAL	
		¢	¢	1	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		¢	¢	0	¢	WEATHER	

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

COORDINATE INVERSES OF LEVEL 2 TO LEVEL 1 TRANSLATION

Table 2
Page 5 of 10

LEVEL 0		LEVEL 1					NAME
COORDINATE	VALUE	TRAJECTORY SETS					
MISSION PROFILE	1	*	*	*	*	AIDS	d
		*	*	*	*	VOR/DME	
		*	*	1	*	AIR DATA	
		0	0	1	*	INERTIAL	
		ø	ø	*	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		ø	ø	*	ø	WEATHER	
MISSION PROFILE	1	*	*	*	*	AIDS	e
		*	*	1	*	VOR/DME	
		*	*	0	*	AIR DATA	
		0	0	1	*	INERTIAL	
		ø	ø	*	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		ø	ø	*	ø	WEATHER	
MISSION PROFILE	1	*	*	*	*	AIDS	f
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		1	*	*	*	INERTIAL	
		ø	ø	*	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		ø	ø	*	ø	WEATHER	
MISSION PROFILE	1	*	*	*	*	AIDS	g
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		0	1	*	*	INERTIAL	
		ø	ø	*	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		ø	ø	*	ø	WEATHER	
MISSION PROFILE	1	*	*	*	*	AIDS	h
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		0	0	0	*	INERTIAL	
		ø	ø	1	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		ø	ø	1	ø	WEATHER	

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

COORDINATE INVERSES OF
LEVEL 2 TO LEVEL 1 TRANSLATION

Table 2
Page 6 of 10

LEVEL 0		LEVEL 1					NAME
COORDINATE	VALUE	TRAJECTORY SETS					
MISSION PROFILE	1	*	*	*	*	AIDS	i
		*	*	0	*	VOR/DME	
		*	*	0	*	AIR DATA	
		0	0	1	*	INERTIAL	
		¢	¢	*	*	AUTOLAND	
		*	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		¢	¢	1	¢	WEATHER	
SAFETY	0	*	*	*	*	AIDS	A
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		0	*	*	*	INERTIAL	
		¢	¢	*	*	AUTOLAND	
		0	0	0	0	ACTIVE FLUTTER CONTROL	
		0	0	0	0	ENGINE CONTROL	
		0	0	0	0	ATTITUDE CONTROL	
		¢	¢	0	¢	WEATHER	
SAFETY	0	*	*	*	*	AIDS	B
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		0	*	*	*	INERTIAL	
		¢	¢	1	*	AUTOLAND	
		0	0	0	0	ACTIVE FLUTTER CONTROL	
		0	0	0	0	ENGINE CONTROL	
		0	0	0	0	ATTITUDE CONTROL	
		¢	¢	1	¢	WEATHER	
SAFETY	0	*	*	*	*	AIDS	C
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		0	*	1	*	INERTIAL	
		¢	¢	0	*	AUTOLAND	
		0	0	0	0	ACTIVE FLUTTER CONTROL	
		0	0	0	0	ENGINE CONTROL	
		0	0	0	0	ATTITUDE CONTROL	
		¢	¢	1	¢	WEATHER	

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

COORDINATE INVERSES OF LEVEL 2 TO LEVEL 1 TRANSLATION

Table 2
Page 7 of 10

LEVEL 0 COORDINATE	VALUE	LEVEL 1 TRAJECTORY SETS					NAME
SAFETY	0	*	*	*	*	AIDS	D
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		0	*	0	0	INERTIAL	
		ø	ø	0	0	AUTOLAND	
		0	0	0	0	ACTIVE FLUTTER CONTROL	
		0	0	0	0	ENGINE CONTROL	
		0	0	0	0	ATTITUDE CONTROL	
		ø	ø	1	ø	WEATHER	
SAFETY	0	*	*	*	*	AIDS	E
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		1	*	*	*	INERTIAL	
		ø	ø	*	*	AUTOLAND	
		0	*	*	*	ACTIVE FLUTTER CONTROL	
		0	*	*	*	ENGINE CONTROL	
		0	*	*	*	ATTITUDE CONTROL	
		ø	ø	*	ø	WEATHER	
SAFETY	1	*	*	*	*	AIDS	f
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		*	*	*	*	INERTIAL	
		ø	ø	*	*	AUTOLAND	
		1	*	*	*	ACTIVE FLUTTER CONTROL	
		*	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		ø	ø	*	ø	WEATHER	
SAFETY	1	*	*	*	*	AIDS	g
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		*	*	*	*	INERTIAL	
		ø	ø	*	*	AUTOLAND	
		0	*	*	*	ACTIVE FLUTTER CONTROL	
		1	*	*	*	ENGINE CONTROL	
		*	*	*	*	ATTITUDE CONTROL	
		ø	ø	*	ø	WEATHER	

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

COORDINATE INVERSES OF LEVEL 2 TO LEVEL 1 TRANSLATION

Table 2
Page 8 of 10

LEVEL 0		LEVEL 1					NAME
COORDINATE	VALUE	TRAJECTORY SETS					
		*	*	*	*	AIDS	
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		*	*	*	*	INERTIAL	
SAFETY	1	¢	¢	*	*	AUTOLAND	h
		0	*	*	*	ACTIVE FLUTTER CONTROL	
		0	*	*	*	ENGINE CONTROL	
		1	*	*	*	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	
		*	*	*	*	AIDS	
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		0	*	*	*	INERTIAL	
SAFETY	1	¢	¢	*	*	AUTOLAND	i
		0	1	*	*	ACTIVE FLUTTER CONTROL	
		0	*	*	*	ENGINE CONTROL	
		0	*	*	*	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	
		*	*	*	*	AIDS	
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		0	*	*	*	INERTIAL	
SAFETY	1	¢	¢	*	*	AUTOLAND	j
		0	0	*	*	ACTIVE FLUTTER CONTROL	
		0	1	*	*	ENGINE CONTROL	
		0	*	*	*	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	
		*	*	*	*	AIDS	
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		0	*	*	*	INERTIAL	
SAFETY	1	¢	¢	*	*	AUTOLAND	k
		0	0	*	*	ACTIVE FLUTTER CONTROL	
		0	0	*	*	ENGINE CONTROL	
		0	1	*	*	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	
		*	*	*	*	AIDS	
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		0	*	*	*	INERTIAL	
SAFETY	1	¢	¢	*	*	AUTOLAND	l
		0	0	1	*	ACTIVE FLUTTER CONTROL	
		0	0	*	*	ENGINE CONTROL	
		0	0	*	*	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	

ORIGINAL PAGE IS OF POOR QUALITY

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

COORDINATE INVERSES OF LEVEL 2 TO LEVEL 1 TRANSLATION

Table 2
Page 9 of 10

LEVEL 0		LEVEL 1 TRAJECTORY SETS					NAME
COORDINATE	VALUE						
SAFETY	1	*	*	*	*	AIDS	m
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		0	*	*	*	INERTIAL	
		¢	¢	*	*	AUTOLAND	
		0	0	0	*	ACTIVE FLUTTER CONTROL	
		0	0	1	*	ENGINE CONTROL	
		0	0	*	*	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	
SAFETY	1	*	*	*	*	AIDS	n
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		0	*	*	*	INERTIAL	
		¢	¢	*	*	AUTOLAND	
		0	0	0	*	ACTIVE FLUTTER CONTROL	
		0	0	0	*	ENGINE CONTROL	
		0	0	1	*	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	
SAFETY	1	*	*	*	*	AIDS	o
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		0	*	*	*	INERTIAL	
		¢	¢	*	*	AUTOLAND	
		0	0	0	1	ACTIVE FLUTTER CONTROL	
		0	0	0	*	ENGINE CONTROL	
		0	0	0	*	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	
SAFETY	1	*	*	*	*	AIDS	p
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		0	*	*	*	INERTIAL	
		¢	¢	*	*	AUTOLAND	
		0	0	0	0	ACTIVE FLUTTER CONTROL	
		0	0	0	1	ENGINE CONTROL	
		0	0	0	*	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	
SAFETY	1	*	*	*	*	AIDS	q
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		0	*	*	*	INERTIAL	
		¢	¢	*	*	AUTOLAND	
		0	0	0	0	ACTIVE FLUTTER CONTROL	
		0	0	0	0	ENGINE CONTROL	
		0	0	0	1	ATTITUDE CONTROL	
		¢	¢	*	¢	WEATHER	

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

COORDINATE INVERSES OF LEVEL 2 TO LEVEL 1 TRANSLATION

Table 2 Page10 of 10

LEVEL 0 COORDINATE	LEVEL 0 VALUE	LEVEL 1 TRAJECTORY SETS					NAME
		*	*	*	*	AIDS	
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		0	*	0	*	INERTIAL	
SAFETY	1	∅	∅	0	1	AUTOLAND	r
		0	0	0	0	ACTIVE FLUTTER CONTROL	
		0	0	0	0	ENGINE CONTROL	
		0	0	0	0	ATTITUDE CONTROL	
		∅	∅	1	∅	WEATHER	
		*	*	*	*	AIDS	
		*	*	*	*	VOR/DME	
		*	*	*	*	AIR DATA	
		0	*	0	1	INERTIAL	
SAFETY	1	∅	∅	0	0	AUTOLAND	s
		0	0	0	0	ACTIVE FLUTTER CONTROL	
		0	0	0	0	ENGINE CONTROL	
		0	0	0	0	ATTITUDE CONTROL	
		∅	∅	1	∅	WEATHER	

ORIGINAL PAGE IS OF POOR QUALITY

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

$$\left[\begin{array}{cccc} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ 0 & * & * & * \\ \not{c} & \not{c} & * & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \not{c} & \not{c} & 0 & \not{c} \end{array}\right] \cup \left[\begin{array}{cccc} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ 0 & * & * & * \\ \not{c} & \not{c} & 1 & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \not{c} & \not{c} & 1 & \not{c} \end{array}\right] \cup \left[\begin{array}{cccc} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ 0 & * & 1 & * \\ \not{c} & \not{c} & 0 & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \not{c} & \not{c} & 1 & \not{c} \end{array}\right] \begin{array}{l} \text{AIDS} \\ \text{VOR/DME} \\ \text{AIR DATA} \\ \text{INERTIAL} \\ \text{AUTOLAND} \\ \text{ACTIVE FLUTTER CONTROL} \\ \text{ENGINE CONTROL} \\ \text{ATTITUDE CONTROL} \\ \text{WEATHER} \end{array}$$

$$\cup \left[\begin{array}{cccc} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ \not{c} & \not{c} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \not{c} & \not{c} & 1 & \not{c} \end{array}\right] \cup \left[\begin{array}{cccc} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ 1 & * & * & * \\ \not{c} & \not{c} & * & * \\ 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \\ \not{c} & \not{c} & \not{c} & \not{c} \end{array}\right] \begin{array}{l} \text{AIDS} \\ \text{VOR/DME} \\ \text{AIR DATA} \\ \text{INERTIAL} \\ \text{AUTOLAND} \\ \text{ACTIVE FLUTTER CONTROL} \\ \text{ENGINE CONTROL} \\ \text{ATTITUDE CONTROL} \\ \text{WEATHER} . \end{array}$$

Finally, the fourth (rightmost) column of the table assigns a single letter name to each level 1 trajectory set; capital letters denote trajectory sets corresponding to coordinate values of 1, while lowercase letters denote trajectory sets associated with coordinate values of 0. Sets are referred to by these names in a later table (Table 3). Thus $(\xi_3\kappa_1)^{-1}(0)$ illustrated above is abbreviated

$$A \cup B \cup C \cup D \cup E.$$

Thus, any aircraft level trajectory which appears in the above set results in a safe mission, and conversely, if a trajectory does not appear in the above set, then the corresponding mission is unsafe.

The next step (in the algorithm for determining γ^{-1}) is to determine the inverse γ_1^{-1} of the level 1 based capability function γ_1 (see [3], p. 26), where, for each level a in the accomplishment set

$$\gamma_1^{-1}(a) = \bigcup_{u \varepsilon \gamma_0^{-1}(a)} \kappa_1^{-1}(u)$$

(In general, the determination of γ_i^{-1} involves "basic variables" as well as "composite variables"; see [3], p. 26. However, in the above case, the only variables at level 0 are composite, and hence they need not be carried down to level 1.) In order to manipulate sets of trajectories (and particularly Cartesian sets), the above formula can be generalized as follows. Suppose that $\gamma_0^{-1}(a)$ is decomposed into a union of Cartesian sets $U_1, U_2, \ldots, U_m$. Then, from the above formula, it follows that

$$\gamma_1^{-1}(a) = \bigcup_{k=1}^{m} \kappa_1^{-1}(U_k). \qquad (3.3.1)$$

Moreover, since U_k is Cartesian, membership of a trajectory u in U_k is uniquely determined by its coordinate memberships, that is,

$$u \varepsilon U_k$$

if and only if, for all coordinates i, $\xi_i(u) \varepsilon \xi_i(U_k)$. This important property says that Cartesian sets of trajectories can be manipulated in a manner similar to that of individual trajectories. In particular, if C is the set of coordinate indices of U^0, then

$$\kappa_1^{-1}(U_k) = \bigcap_{k=1} (\xi_i \kappa_i)^{-1}(\xi_i(U_k)). \qquad (3.3.2)$$

Applying these formulas to the computation in question, we note first that each $\gamma_0^{-1}(a)$ is already Cartesian. (See Table 1.) Thus a decomposition of $\gamma_0^{-1}(a)$ into Cartesian sets is trivial, i.e., $\gamma_0^{-1}(a) = U_1$ and, by equation (3.3.1), we have

$$\gamma_1^{-1}(a) = \kappa_1^{-1}(U_1).$$

Next, we note that $C=\{1,2,3,4\}$ and hence, by equation(3.3.2),

$$\gamma_1^{-1}(a) = \bigcap_{i=1}^{4} (\xi_i\kappa_1)^{-1}(\xi_i(U_1)). \qquad (3.3.3)$$

The values of the intersected terms on the right are determined using Table 2 such that each term is expressed as a union of Cartesian sets. (Note that if $\xi_i(U_1) = * = \xi_i(U^0)$ (an arbitrary value) or if $\xi_i(U_1) = \not{c}$ (the i^{th} coordinate of U^0 is a constant) then the term $(\xi_i\kappa_1)^{-1}(\xi_i(U_1))$ can be ignored, since it is equal to the whole level 1 trajectory space U^1.) Finally, these unions are intersected according to (3.3.3), the result being an expression of $\gamma_1^{-1}(a)$ as a union of level 1 Cartesian trajectory sets. These resulting sets are displayed in Table 3. To illustrate this computation and to aid the interpretation of Table 3, consider the following example.

Example

Suppose $a=a_0$. Then, by row a(0) of Table 1,

$$\gamma^{-1}(a_0) = U_1 = \begin{array}{l} \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array}\right] \end{array} \begin{array}{l} \text{ECONOMICS} \\ \text{OPERATIONS} \\ \text{PROFILE} \\ \text{SAFETY.} \end{array}$$

Thus,

$$\xi_1(U_1) = 0,$$
$$\xi_2(U_1) = 0,$$
$$\xi_3(U_1) = 0,$$
$$\xi_3(U_1) = 0,$$

where the correspondence between coordinate indices and coordinate names ([illegible] Table 2) is:

LEVEL 1 BASED
CAPABILITY FUNCTION

Table 3
Page 1 of 22

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS	
	ECONOMICS	OPERATIONS	PROFILE	SAFETY		
					0 0 0 0	AIDS
					0 0 0 0	VOR/DME
					0 0 0 0	AIR DATA
					0 0 0 *	INERTIAL
a(0)	A	A	A	A	¢ ¢ 0 *	AUTOLAND
					0 0 0 0	ACTIVE FLUTTER CONTROL
					0 0 0 0	ENGINE CONTROL
					0 0 0 0	ATTITUDE CONTROL
					¢ ¢ 0 ¢	WEATHER
a(0)	A	A	A	B	Ø	(NULL ARRAY)
a(0)	A	A	A	C	Ø	(NULL ARRAY)
					0 0 0 0	AIDS
					0 0 0 0	VOR/DME
					0 0 0 0	AIR DATA
					0 0 0 0	INERTIAL
a(0)	A	A	A	D	¢ ¢ 0 0	AUTOLAND
					0 0 0 0	ACTIVE FLUTTER CONTROL
					0 0 0 0	ENGINE CONTROL
					0 0 0 0	ATTITUDE CONTROL
					¢ ¢ 1 ¢	WEATHER
a(0)	A	A	A	E	Ø	(NULL ARRAY)
					0 0 0 0	AIDS
					0 0 0 0	VOR/DME
					0 0 0 0	AIR DATA
					0 0 1 *	INERTIAL
a(0)	A	A	B	A	¢ ¢ * *	AUTOLAND
					0 0 0 0	ACTIVE FLUTTER CONTROL
					0 0 0 0	ENGINE CONTROL
					0 0 0 0	ATTITUDE CONTROL
					¢ ¢ 0 ¢	WEATHER
a(0)	A	A	B	B	Ø	(NULL ARRAY)
a(0)	A	A	B	C	Ø	(NULL ARRAY)
a(0)	A	A	B	D	Ø	(NULL ARRAY)
a(0)	A	A	B	E	Ø	(NULL ARRAY)
					0 0 0 0	AIDS
					0 0 0 0	VOR/DME
					0 0 0 0	AIR DATA
					0 0 0 *	INERTIAL
a(0)	A	A	C	A	¢ ¢ 1 *	AUTOLAND
					0 0 0 0	ACTIVE FLUTTER CONTROL
					0 0 0 0	ENGINE CONTROL
					0 0 0 0	ATTITUDE CONTROL
					¢ ¢ 0 ¢	WEATHER

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

Names are defined in Table 2.

For each row, the resultant level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 1 BASED
CAPABILITY FUNCTION

Table 3
Page 2 of 22

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS
	ECONOMICS	OPERATIONS	PROFILE	SAFETY	
a(0)	A	A	C	B	∅ (NULL ARRAY)
a(0)	A	A	C	C	∅ (NULL ARRAY)
a(0)	A	A	C	D	∅ (NULL ARRAY)
a(0)	A	A	C	E	∅ (NULL ARRAY)
a(1)	b	A	A	A	1 * * * AIDS 0 0 0 0 VOR/DME 0 0 0 0 AIR DATA 0 0 0 * INERTIAL ¢ ¢ 0 * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 0 ¢ WEATHER
a(1)	b	A	A	B	∅ (NULL ARRAY)
a(1)	b	A	A	C	∅ (NULL ARRAY)
a(1)	b	A	A	D	1 * * * AIDS 0 0 0 0 VOR/DME 0 0 0 0 AIR DATA 0 0 0 0 INERTIAL ¢ ¢ 0 0 AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 1 ¢ WEATHER
a(1)	b	A	A	E	∅ (NULL ARRAY)
a(1)	b	A	B	A	1 * * * AIDS 0 0 0 0 VOR/DME 0 0 0 0 AIR DATA 0 0 1 * INERTIAL ¢ ¢ * * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL * * 0 * WEATHER
a(1)	b	A	B	B	∅ (NULL ARRAY)
a(1)	b	A	B	C	∅ (NULL ARRAY)
a(1)	b	A	B	D	∅ (NULL ARRAY)
a(1)	b	A	B	E	∅ (NULL ARRAY)

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

Names are defined in Table 2.

For each row, the resultant level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 1 BASED
CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS
	ECONOMICS	OPERATIONS	PROFILE	SAFETY	
a(1)	b	A	C	A	1 * * * AIDS 0 0 0 0 VOR/DME 0 0 0 0 AIR DATA 0 0 0 * INERTIAL ¢ ¢ 1 * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 0 ¢ WEATHER
a(1)	b	A	C	B	ø (NULL ARRAY)
a(1)	b	A	C	C	ø (NULL ARRAY)
a(1)	b	A	C	D	ø (NULL ARRAY)
a(1)	b	A	C	E	ø (NULL ARRAY)
a(1)	c	A	A	A	0 1 * * AIDS 0 0 0 0 VOR/DME 0 0 0 0 AIR DATA 0 0 0 * INERTIAL ¢ ¢ 0 * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 0 ¢ WEATHER
a(1)	c	A	A	B	ø (NULL ARRAY)
a(1)	c	A	A	C	ø (NULL ARRAY)
a(1)	c	A	A	D	0 1 * * AIDS 0 0 0 0 VOR/DME 0 0 0 0 AIR DATA 0 0 0 0 INERTIAL ¢ ¢ 0 0 AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 1 ¢ WEATHER
a(1)	c	A	A	E	ø (NULL ARRAY)
a(1)	c	A	B	A	0 1 * * AIDS 0 0 0 0 VOR/DME 0 0 0 0 AIR DATA 0 0 1 * INERTIAL ¢ ¢ * * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 0 ¢ WEATHER

Column 1: Phase 1 = Takeoff/Cruise A Column 2: Phase 2 = Cruise B Column 3: Phase 3 = Cruise C Column 4: Phase 4 = Landing	Names are defined in Table 2.	For each row, the resultant level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 1 BASED
CAPABILITY FUNCTION

Table 3
Page 4 of 22

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS
	ECONOMICS	OPERATIONS	PROFILE	SAFETY	
a (1)	c	A	B	B	Ø (NULL ARRAY),
a (1)	c	A	B	C	Ø (NULL ARRAY)
a (1)	c	A	B	D	Ø (NULL ARRAY)
a (1)	c	A	B	E	Ø (NULL ARRAY)
a (1)	c	A	C	A	$\left[\begin{array}{cccc} 0 & 1 & * & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & * \\ \text{ø} & \text{ø} & 1 & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \text{ø} & \text{ø} & 0 & \text{ø} \end{array}\right] \begin{array}{l} \text{AIDS} \\ \text{VOR/DME} \\ \text{AIR DATA} \\ \text{INERTIAL} \\ \text{AUTOLAND} \\ \text{ACTIVE FLUTTER CONTROL} \\ \text{ENGINE CONTROL} \\ \text{ATTITUDE CONTROL} \\ \text{WEATHER} \end{array}$
a (1)	c	A	C	B	Ø (NULL ARRAY)
a (1)	c	A	C	C	Ø (NULL ARRAY)
a (1)	c	A	C	D	Ø (NULL ARRAY)
a (1)	c	A	C	E	Ø (NULL ARRAY)
a (1)	d	A	A	A	$\left[\begin{array}{cccc} 0 & 0 & 1 & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & * \\ \text{ø} & \text{ø} & * & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \text{ø} & \text{ø} & 0 & \text{ø} \end{array}\right] \begin{array}{l} \text{AIDS} \\ \text{VOR/DME} \\ \text{AIR DATA} \\ \text{INERTIAL} \\ \text{AUTOLAND} \\ \text{ACTIVE FLUTTER CONTROL} \\ \text{ENGINE CONTROL} \\ \text{ATTITUDE CONTROL} \\ \text{WEATHER} \end{array}$
a (1)	d	A	A	B	Ø (NULL ARRAY)
a (1)	d	A	A	C	Ø (NULL ARRAY)
a (1)	d	A	A	D	$\left[\begin{array}{cccc} 0 & 0 & 1 & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \text{ø} & \text{ø} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \text{ø} & \text{ø} & 1 & \text{ø} \end{array}\right] \begin{array}{l} \text{AIDS} \\ \text{VOR/DME} \\ \text{AIR DATA} \\ \text{INERTIAL} \\ \text{AUTOLAND} \\ \text{ACTIVE FLUTTER CONTROL} \\ \text{ENGINE CONTROL} \\ \text{ATTITUDE CONTROL} \\ \text{WEATHER} \end{array}$
a (1)	d	A	A	E	Ø (NULL ARRAY)

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

Names are defined in Table 2.

For each row, the resultant level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 1 BASED
CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS	
	ECONOMICS	OPERATIONS	PROFILE	SAFETY		
a(1)	d	A	B	A	0 0 1 *	AIDS
					0 0 0 0	VOR/DME
					0 0 0 0	AIR DATA
					0 0 1 *	INERTIAL
					¢ ¢ * *	AUTOLAND
					0 0 0 0	ACTIVE FLUTTER CONTROL
					0 0 0 0	ENGINE CONTROL
					0 0 0 0	ATTITUDE CONTROL
					¢ ¢ 0 ¢	WEATHER
a(1)	d	A	B	B	∅	(NULL ARRAY)
a(1)	d	A	B	C	∅	(NULL ARRAY)
a(1)	d	A	B	D	∅	(NULL ARRAY)
a(1)	d	A	B	E	∅	(NULL ARRAY)
a(1)	d	A	C	A	0 0 1 *	AIDS
					0 0 0 0	VOR/DME
					0 0 0 0	AIR DATA
					0 0 0 *	INERTIAL
					¢ ¢ 1 *	AUTOLAND
					0 0 0 0	ACTIVE FLUTTER CONTROL
					0 0 0 0	ENGINE CONTROL
					0 0 0 0	ATTITUDE CONTROL
					¢ ¢ 0 ¢	WEATHER
a(1)	d	A	C	B	∅	(NULL ARRAY)
a(1)	d	A	C	C	∅	(NULL ARRAY)
a(1)	d	A	C	D	∅	(NULL ARRAY)
a(1)	d	A	C	E	∅	(NULL ARRAY)
a(1)	e	A	A	A	0 0 0 1	AIDS
					0 0 0 0	VOR/DME
					0 0 0 0	AIR DATA
					0 0 0 *	INERTIAL
					¢ ¢ 0 *	AUTOLAND
					0 0 0 0	ACTIVE FLUTTER CONTROL
					0 0 0 0	ENGINE CONTROL
					0 0 0 0	ATTITUDE CONTROL
					¢ ¢ 0 ¢	WEATHER
a(1)	e	A	A	B	∅	(NULL ARRAY)
a(1)	e	A	A	C	∅	(NULL ARRAY)

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

Names are defined in Table 2.

For each row, the resultant level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 1 BASED
CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS	
	ECONOMICS	OPERATIONS	PROFILE	SAFETY		
					0 0 0 1	AIDS
					0 0 0 0	VOR/DME
					0 0 0 0	AIR DATA
					0 0 0 0	INERTIAL
a(1)	e	A	A	D	¢ ¢ 0 0	AUTOLAND
					0 0 0 0	ACTIVE FLUTTER CONTROL
					0 0 0 0	ENGINE CONTROL
					0 0 0 0	ATTITUDE CONTROL
					¢ ¢ 1 ¢	WEATHER
a(1)	e	A	A	E	Ø	(NULL ARRAY)
					0 0 0 1	AIDS
					0 0 0 0	VOR/DME
					0 0 0 0	AIR DATA
					0 0 1 *	INERTIAL
a(1)	e	A	B	A	¢ ¢ * *	AUTOLAND
					0 0 0 0	ACTIVE FLUTTER CONTROL
					0 0 0 0	ENGINE CONTROL
					0 0 0 0	ATTITUDE CONTROL
					¢ ¢ 0 ¢	WEATHER
a(1)	e	A	B	B	Ø	(NULL ARRAY)
a(1)	e	A	B	C	Ø	(NULL ARRAY)
a(1)	e	A	B	D	Ø	(NULL ARRAY)
a(1)	e	A	B	E	Ø	(NULL ARRAY)
					0 0 0 1	AIDS
					0 0 0 0	VOR/DME
					0 0 0 0	AIR DATA
					0 0 0 *	INERTIAL
a(1)	e	A	C	A	¢ ¢ 1 *	AUTOLAND
					0 0 0 0	ACTIVE FLUTTER CONTROL
					0 0 0 0	ENGINE CONTROL
					0 0 0 0	ATTITUDE CONTROL
					¢ ¢ 0 ¢	WEATHER
a(1)	e	A	C	B	Ø	(NULL ARRAY)
a(1)	e	A	C	C	Ø	(NULL ARRAY)
a(1)	e	A	C	D	Ø	(NULL ARRAY)
a(1)	e	A	C	E	Ø	(NULL ARRAY)

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

Names are defined in Table 2.

For each row, the resultant level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 1 BASED
CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS				
	ECONOMICS	OPERATIONS	PROFILE	SAFETY					
					*	*	*	*	AIDS
					1	*	*	*	VOR/DME
					*	*	*	*	AIR DATA
					0	0	0	*	INERTIAL
a(2)	*	b	A	A	¢	¢	0	*	AUTOLAND
					0	0	0	0	ACTIVE FLUTTER CONTROL
					0	0	0	0	ENGINE CONTROL
					0	0	0	0	ATTITUDE CONTROL
					¢	¢	0	¢	WEATHER
a(2)	*	A	A	B				∅	(NULL ARRAY)
a(2)	*	A	A	C				∅	(NULL ARRAY)
					*	*	*	*	AIDS
					1	*	*	*	VOR/DME
					*	*	*	*	AIR DATA
					0	0	0	0	INERTIAL
a(2)	*	b	A	D	¢	¢	0	0	AUTOLAND
					0	0	0	0	ACTIVE FLUTTER CONTROL
					0	0	0	0	ENGINE CONTROL
					0	0	0	0	ATTITUDE CONTROL
					¢	¢	1	¢	WEATHER
a(2)	*	b	A	E				∅	(NULL ARRAY)
					*	*	*	*	AIDS
					1	*	0	*	VOR/DME
					*	*	0	*	AIR DATA
					0	0	1	*	INERTIAL
a(2)	*	b	B	A	¢	¢	*	*	AUTOLAND
					0	0	0	0	ACTIVE FLUTTER CONTROL
					0	0	0	0	ENGINE CONTROL
					0	0	0	0	ATTITUDE CONTROL
					¢	¢	0	¢	WEATHER
a(2)	*	b	B	B				∅	(NULL ARRAY)
a(2)	*	b	B	C				∅	(NULL ARRAY)
a(2)	*	b	B	D				∅	(NULL ARRAY)
a(2)	*	b	B	E				∅	(NULL ARRAY)
					*	*	*	*	AIDS
					1	*	*	*	VOR/DME
					*	*	*	*	AIR DATA
					0	0	0	*	INERTIAL
a(2)	*	b	C	A	¢	¢	1	*	AUTOLAND
					0	0	0	0	ACTIVE FLUTTER CONTROL
					0	0	0	0	ENGINE CONTROL
					0	0	0	0	ATTITUDE CONTROL
					¢	¢	0	¢	WEATHER

Column 1: Phase 1 = Takeoff/Cruise A Column 2: Phase 2 = Cruise B Column 3: Phase 3 = Cruise C Column 4: Phase 4 = Landing	Names are defined in Table 2.	For each row, the resultant level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 1 BASED
CAPABILITY FUNCTION

Table 3
Page 8 of 22

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS
	ECONOMICS	OPERATIONS	PROFILE	SAFETY	
a(2)	*	b	C	B	∅ (NULL ARRAY)
a(2)	*	b	C	C	∅ (NULL ARRAY)
a(2)	*	b	C	D	∅ (NULL ARRAY)
a(2)	*	b	C	E	∅ (NULL ARRAY)
a(2)	*	c	A	A	* * * * AIDS 0 1 * * VOR/DME * * * * AIR DATA 0 0 0 * INERTIAL ¢ ¢ 0 * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 0 ¢ WEATHER
a(2)	*	c	A	B	∅ (NULL ARRAY)
a(2)	*	c	A	C	∅ (NULL ARRAY)
a(2)	*	c	A	D	* * * * AIDS 0 1 * * VOR/DME * * * * AIR DATA 0 0 0 0 INERTIAL ¢ ¢ 0 0 AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 1 ¢ WEATHER
a(2)	*	c	A	E	∅ (NULL ARRAY)
a(2)	*	c	B	A	* * * * AIDS 0 1 0 * VOR/DME * * 0 * AIR DATA 0 0 1 * INERTIAL ¢ ¢ * * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 0 ¢ WEATHER
a(2)	*	c	B	B	∅ (NULL ARRAY)
a(2)	*	c	B	C	∅ (NULL ARRAY)
a(2)	*	c	B	D	∅ (NULL ARRAY)
a(2)	*	c	B	E	∅ (NULL ARRAY)

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

Names are defined in Table 2.

For each row, the resultant level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 1 BASED
CAPABILITY FUNCTION

Table 3
Page 9 of 22

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS
	ECONOMICS	OPERATIONS	PROFILE	SAFETY	
a (2)	*	c	C	A	* * * * AIDS 0 1 * * VOR/DME * * * * AIR DATA 0 0 0 * INERTIAL ¢ ¢ 1 * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 0 ¢ WEATHER
a (2)	*	c	C	B	Ø (NULL ARRAY)
a (2)	*	c	C	C	Ø (NULL ARRAY)
a (2)	*	c	C	D	Ø (NULL ARRAY)
a (2)	*	c	C	E	Ø (NULL ARRAY)
a (2)	*	d	A	A	* * * * AIDS 0 0 1 * VOR/DME * * * * AIR DATA 0 0 0 * INERTIAL ¢ ¢ 0 * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 0 ¢ WEATHER
a (2)	*	d	A	B	Ø (NULL ARRAY)
a (2)	*	d	A	C	Ø (NULL ARRAY)
a (2)	*	d	A	D	* * * * AIDS 0 0 1 * VOR/DME * * * * AIR DATA 0 0 0 0 INERTIAL ¢ ¢ 0 0 AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 1 ¢ WEATHER
a (2)	*	d	A	E	Ø (NULL ARRAY)
a (2)	*	d	B	A	Ø (NULL ARRAY)
a (2)	*	d	B	B	Ø (NULL ARRAY)
a (2)	*	d	B	C	Ø (NULL ARRAY)
a (2)	*	d	B	D	Ø (NULL ARRAY)
a (2)	*	d	B	E	Ø (NULL ARRAY)

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

Names are defined in Table 2.

For each row, the resultant level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 1 BASED CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS
	ECONOMICS	OPERATIONS	PROFILE	SAFETY	
a(2)	*	d	C	A	$\left[\begin{array}{cccc} * & * & * & * \\ 0 & 0 & 1 & * \\ * & * & * & * \\ 0 & 0 & 0 & * \\ \text{¢} & \text{¢} & 1 & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \text{¢} & \text{¢} & 0 & \text{¢} \end{array}\right] \begin{array}{l} \text{AIDS} \\ \text{VOR/DME} \\ \text{AIR DATA} \\ \text{INERTIAL} \\ \text{AUTOLAND} \\ \text{ACTIVE FLUTTER CONTROL} \\ \text{ENGINE CONTROL} \\ \text{ATTITUDE CONTROL} \\ \text{WEATHER} \end{array}$
a(2)	*	d	C	B	Ø (NULL ARRAY)
a(2)	*	d	C	C	Ø (NULL ARRAY)
a(2)	*	d	C	D	Ø (NULL ARRAY)
a(2)	*	d	C	E	Ø (NULL ARRAY)
					Ø (NULL ARRAY)
a(2)	*	e	A	A	$\left[\begin{array}{cccc} * & * & * & * \\ 0 & 0 & 0 & 1 \\ * & * & * & * \\ 0 & 0 & 0 & * \\ \text{¢} & \text{¢} & 0 & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \text{¢} & \text{¢} & 0 & \text{¢} \end{array}\right] \begin{array}{l} \text{AIDS} \\ \text{VOR/DME} \\ \text{AIR DATA} \\ \text{INERTIAL} \\ \text{AUTOLAND} \\ \text{ACTIVE FLUTTER CONTROL} \\ \text{ENGINE CONTROL} \\ \text{ATTITUDE CONTROL} \\ \text{WEATHER} \end{array}$
a(2)	*	e	A	B	Ø (NULL ARRAY)
a(2)	*	e	A	C	Ø (NULL ARRAY)
a(2)	*	e	A	D	$\left[\begin{array}{cccc} * & * & * & * \\ 0 & 0 & 0 & 1 \\ * & * & * & * \\ 0 & 0 & 0 & 0 \\ \text{¢} & \text{¢} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \text{¢} & \text{¢} & 1 & \text{¢} \end{array}\right] \begin{array}{l} \text{AIDS} \\ \text{VOR/DME} \\ \text{AIR DATA} \\ \text{INERTIAL} \\ \text{AUTOLAND} \\ \text{ACTIVE FLUTTER CONTROL} \\ \text{ENGINE CONTROL} \\ \text{ATTITUDE CONTROL} \\ \text{WEATHER} \end{array}$
a(2)	*	e	A	E	Ø (NULL ARRAY)
a(2)	*	e	B	A	$\left[\begin{array}{cccc} * & * & * & * \\ 0 & 0 & 0 & 1 \\ * & * & 0 & * \\ 0 & 0 & 1 & * \\ \text{¢} & \text{¢} & * & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \text{¢} & \text{¢} & 0 & \text{¢} \end{array}\right] \begin{array}{l} \text{AIDS} \\ \text{VOR/DME} \\ \text{AIR DATA} \\ \text{INERTIAL} \\ \text{AUTOLAND} \\ \text{ACTIVE FLUTTER CONTROL} \\ \text{ENGINE CONTROL} \\ \text{ATTITUDE CONTROL} \\ \text{WEATHER} \end{array}$

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

Names are defined in Table 2.

For each row, the resultant level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 1 BASED
CAPABILITY FUNCTION

Table 3
Page 11 of 22

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS
	ECONOMICS	OPERATIONS	PROFILE	SAFETY	
a(2)	*	e	B	B	ø (NULL ARRAY)
a(2)	*	e	B	C	ø (NULL ARRAY)
a(2)	*	e	B	D	ø (NULL ARRAY)
a(2)	*	e	B	E	ø (NULL ARRAY)
a(2)	*	e	C	A	* * * * AIDS 0 0 0 1 VOR/DME * * * * AIR DATA 0 0 0 * INERTIAL ¢ ¢ 1 * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 0 ¢ WEATHER
a(2)	*	e	C	B	ø (NULL ARRAY)
a(2)	*	e	C	C	ø (NULL ARRAY)
a(2)	*	e	C	D	ø (NULL ARRAY)
a(2)	*	e	C	E	ø (NULL ARRAY)
a(2)	*	f	A	A	* * * * AIDS 0 0 0 0 VOR/DME 1 * * * AIR DATA 0 0 0 * INERTIAL ¢ ¢ 0 * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 0 ¢ WEATHER
a(2)	*	f	A	B	ø (NULL ARRAY)
a(2)	*	f	A	C	ø (NULL ARRAY)
a(2)	*	f	A	D	* * * * AIDS 0 0 0 0 VOR/DME 1 * * * AIR DATA 0 0 0 0 INERTIAL ¢ ¢ 0 0 AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 1 ¢ WEATHER
a(2)	*	f	A	E	ø (NULL ARRAY)

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

Names are defined in Table 2.

For each row, the resultant level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 1 BASED
CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS ECONOMICS	 OPERATIONS	 PROFILE	 SAFETY	RESULTING LEVEL 1 TRAJECTORY SETS
a(2)	*	f	B	A	* * * * AIDS 0 0 0 0 VOR/DME 1 * 0 * AIR DATA 0 0 1 * INERTIAL ɇ ɇ * * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ɇ ɇ 0 ɇ WEATHER
a(2)	*	f	B	B	Ø (NULL ARRAY)
a(2)	*	f	B	C	Ø (NULL ARRAY)
a(2)	*	f	B	D	Ø (NULL ARRAY)
a(2)	*	f	B	E	Ø (NULL ARRAY)
a(2)	*	f	C	A	* * * * AIDS 0 0 0 0 VOR/DME 1 * * * AIR DATA 0 0 0 * INERTIAL ɇ ɇ 1 * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ɇ ɇ 0 ɇ WEATHER
a(2)	*	f	C	B	Ø (NULL ARRAY)
a(2)	*	f	C	C	Ø (NULL ARRAY)
a(2)	*	f	C	D	Ø (NULL ARRAY)
a(2)	*	f	C	E	Ø (NULL ARRAY)
a(2)	*	g	A	A	* * * * AIDS 0 0 0 0 VOR/DME 0 1 * * AIR DATA 0 0 0 * INERTIAL ɇ ɇ 0 * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ɇ ɇ 0 ɇ WEATHER
a(2)	*	g	A	B	Ø (NULL ARRAY)
a(2)	*	g	A	C	Ø (NULL ARRAY)

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

Names are defined in Table 2.

For each row, the resultant level 1 trajectory set is the intersection of the sets named in Column 2.

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS
	ECONOMICS	OPERATIONS	PROFILE	SAFETY	
a(2)	*	g	A	D	* * * * AIDS 0 0 0 0 VOR/DME 0 1 * * AIR DATA 0 0 0 0 INERTIAL ø ø 0 0 AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ø ø 1 ø WEATHER
a(2)	*	g	A	E	Ø (NULL ARRAY)
a(2)	*	g	B	A	* * * * AIDS 0 0 0 0 VOR/DME 0 1 0 * AIR DATA 0 0 1 * INERTIAL ø ø * * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ø ø 0 ø WEATHER
a(2)	*	g	B	B	Ø (NULL ARRAY)
a(2)	*	g	B	C	Ø (NULL ARRAY)
a(2)	*	g	B	D	Ø (NULL ARRAY)
a(2)	*	g	B	E	Ø (NULL ARRAY)
a(2)	*	g	C	A	* * * * AIDS 0 0 0 0 VOR/DME 0 1 * * AIR DATA 0 0 0 * INERTIAL ø ø 1 * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ø ø 0 ø WEATHER
a(2)	*	g	C	B	Ø (NULL ARRAY)
a(2)	*	g	C	C	Ø (NULL ARRAY)
a(2)	*	g	C	D	Ø (NULL ARRAY)
a(2)	*	g	C	E	Ø (NULL ARRAY)

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

Names are defined in Table 2.

For each row, the resultant level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 1 BASED CAPABILITY FUNCTION

Table 3
Page 14 of 22

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS
	ECONOMICS	OPERATIONS	PROFILE	SAFETY	
a(2)	*	h	A	A	* * * * AIDS 0 0 0 0 VOR/DME 0 0 1 * AIR DATA 0 0 0 * INERTIAL ¢ ¢ 0 * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 0 ¢ WEATHER
a(2)	*	h	A	B	ø (NULL ARRAY)
a(2)	*	h	A	C	ø (NULL ARRAY)
a(2)	*	h	A	D	* * * * AIDS 0 0 0 0 VOR/DME 0 0 1 * AIR DATA 0 0 0 0 INERTIAL ¢ ¢ 0 0 AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 1 ¢ WEATHER
a(2)	*	h	A	E	ø (NULL ARRAY)
a(2)	*	h	B	A	ø (NULL ARRAY)
a(2)	*	h	B	B	ø (NULL ARRAY)
a(2)	*	h	B	C	ø (NULL ARRAY)
a(2)	*	h	B	D	ø (NULL ARRAY)
a(2)	*	h	B	E	ø (NULL ARRAY)
a(2)	*	h	C	A	* * * * AIDS 0 0 0 0 VOR/DME 0 0 1 * AIR DATA 0 0 0 * INERTIAL ¢ ¢ 1 * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 0 ¢ WEATHER
a(2)	*	h	C	B	ø (NULL ARRAY)
a(2)	*	h	C	C	ø (NULL ARRAY)
a(2)	*	h	C	D	ø (NULL ARRAY)
a(2)	*	h	C	E	ø (NULL ARRAY)

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

Names are defined in Table 2.

For each row, the resultant level 1 trajectory set is the intersection of the sets named in Column 2.

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS
	ECONOMICS	OPERATIONS	PROFILE	SAFETY	
a(2)	*	i	A	A	$\left[\begin{array}{cccc} * & * & * & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & * \\ \text{¢} & \text{¢} & 0 & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \text{¢} & \text{¢} & 0 & \text{¢} \end{array}\right] \begin{array}{l} \text{AIDS} \\ \text{VOR/DME} \\ \text{AIR DATA} \\ \text{INERTIAL} \\ \text{AUTOLAND} \\ \text{ACTIVE FLUTTER CONTROL} \\ \text{ENGINE CONTROL} \\ \text{ATTITUDE CONTROL} \\ \text{WEATHER} \end{array}$
a(2)	*	i	A	B	Ø (NULL ARRAY)
a(2)	*	i	A	C	Ø (NULL ARRAY)
a(2)	*	i	A	D	$\left[\begin{array}{cccc} * & * & * & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ \text{¢} & \text{¢} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \text{¢} & \text{¢} & 1 & \text{¢} \end{array}\right] \begin{array}{l} \text{AIDS} \\ \text{VOR/DME} \\ \text{AIR DATA} \\ \text{INERTIAL} \\ \text{AUTOLAND} \\ \text{ACTIVE FLUTTER CONTROL} \\ \text{ENGINE CONTROL} \\ \text{ATTITUDE CONTROL} \\ \text{WEATHER} \end{array}$
a(2)	*	i	A	E	Ø (NULL ARRAY)
a(2)	*	i	B	A	$\left[\begin{array}{cccc} * & * & * & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & * \\ \text{¢} & \text{¢} & * & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \text{¢} & \text{¢} & 0 & \text{¢} \end{array}\right] \begin{array}{l} \text{AIDS} \\ \text{VOR/DME} \\ \text{AIR DATA} \\ \text{INERTIAL} \\ \text{AUTOLAND} \\ \text{ACTIVE FLUTTER CONTROL} \\ \text{ENGINE CONTROL} \\ \text{ATTITUDE CONTROL} \\ \text{WEATHER} \end{array}$
a(2)	*	i	B	B	Ø (NULL ARRAY)
a(2)	*	i	B	C	Ø (NULL ARRAY)
a(2)	*	i	B	D	Ø (NULL ARRAY)
a(2)	*	i	B	E	Ø (NULL ARRAY)
a(2)	*	i	C	A	$\left[\begin{array}{cccc} * & * & * & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & * \\ \text{¢} & \text{¢} & 1 & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \text{¢} & \text{¢} & 0 & \text{¢} \end{array}\right] \begin{array}{l} \text{AIDS} \\ \text{VOR/DME} \\ \text{AIR DATA} \\ \text{INERTIAL} \\ \text{AUTOLAND} \\ \text{ACTIVE FLUTTER CONTROL} \\ \text{ENGINE CONTROL} \\ \text{ATTITUDE CONTROL} \\ \text{WEATHER} \end{array}$

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

Names are defined in Table 2.

For each row, the resultant level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 1 BASED
CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS
	ECONOMICS	OPERATIONS	PROFILE	SAFETY	
a(2)	*	i	c	B	Ø (NULL ARRAY)
a(2)	*	i	c	C	Ø (NULL ARRAY)
a(2)	*	i	c	D	Ø (NULL ARRAY)
a(2)	*	i	c	E	Ø (NULL ARRAY)
a(3)	*	*	d	A	* * * * AIDS * * * * VOR/DME * * 1 * AIR DATA 0 0 1 * INERTIAL ¢ ¢ * * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 0 ¢ WEATHER
a(3)	*	*	d	B	* * * * AIDS * * * * VOR/DME * * 1 * AIR DATA 0 0 1 * INERTIAL ¢ ¢ 1 * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 1 ¢ WEATHER
a(3)	*	*	d	C	* * * * AIDS * * * * VOR/DME * * 1 * AIR DATA 0 0 1 * INERTIAL ¢ ¢ 0 * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 1 ¢ WEATHER
a(3)	*	*	d	D	Ø (NULL ARRAY)
a(3)	*	*	d	E	Ø (NULL ARRAY)
a(3)	*	*	e	A	* * * * AIDS * * 1 * VOR/DME * * 0 * AIR DATA 0 0 1 * INERTIAL ¢ ¢ * * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 0 ¢ WEATHER

Column 1: [illegible] are defined [illegible] Table [illegible] Column 2: [illegible]

For each row, the resultant Level 1 trajectory set is [illegible] intersection of the sets [illegible] in Column 2.

LEVEL 1 BASED
CAPABILITY FUNCTION

Table 3
Page 17 of 22

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS
	ECONOMICS	OPERATIONS	PROFILE	SAFETY	
a (3)	*	*	e	B	* * * * AIDS * * 1 * VOR/DME * * 0 * AIR DATA 0 0 1 * INERTIAL ɇ ɇ 1 * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ɇ ɇ 1 ɇ WEATHER
a (3)	*	*	e	C	* * * * AIDS * * 1 * VOR/DME * * 0 * AIR DATA 0 0 1 * INERTIAL ɇ ɇ 0 * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ɇ ɇ 1 ɇ WEATHER
a (3)	*	*	e	D	∅ (NULL ARRAY)
a (3)	*	*	e	E	∅ (NULL ARRAY)
a (3)	*	*	f	A	∅ (NULL ARRAY)
a (3)	*	*	f	B	∅ (NULL ARRAY)
a (3)	*	*	f	C	∅ (NULL ARRAY)
a (3)	*	*	f	D	∅ (NULL ARRAY)
a (3)	*	*	f	E	* * * * AIDS * * * * VOR/DME * * * * AIR DATA 1 * * * INERTIAL ɇ ɇ * * AUTOLAND 0 * * * ACTIVE FLUTTER CONTROL 0 * * * ENGINE CONTROL 0 * * * ATTITUDE CONTROL ɇ ɇ * ɇ WEATHER
a (3)	*	*	g	A	* * * * AIDS * * * * VOR/DME * * * * AIR DATA 0 1 * * INERTIAL ɇ ɇ * * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ɇ ɇ 0 ɇ WEATHER

Column notes	Names	Resulting sets
Column 1: Phase 1 = Takeoff/Cruise A Column 2: Phase 2 = Cruise B Column 3: Phase 3 = Cruise C Column 4: Phase 4 = Landing	Names are defined in Table 2.	For each row, the resultant Level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 1 BASED CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS				
	ECONOMICS	OPERATIONS	PROFILE	SAFETY					
a(3)	*	*	g	B	*	*	*	*	AIDS
					*	*	*	*	VOR/DME
					*	*	*	*	AIR DATA
					0	1	*	*	INERTIAL
					¢	¢	1	*	AUTOLAND
					0	0	0	0	ACTIVE FLUTTER CONTROL
					0	0	0	0	ENGINE CONTROL
					0	0	0	0	ATTITUDE CONTROL
					¢	¢	1	¢	WEATHER
a(3)	*	*	g	C	*	*	*	*	AIDS
					*	*	*	*	VOR/DME
					*	*	*	*	AIR DATA
					0	1	1	*	INERTIAL
					¢	¢	0	*	AUTOLAND
					0	0	0	0	ACTIVE FLUTTER CONTROL
					0	0	0	0	ENGINE CONTROL
					0	0	0	0	ATTITUDE CONTROL
					¢	¢	1	¢	WEATHER
a(3)	*	*	g	D	*	*	*	*	AIDS
					*	*	*	*	VOR/DME
					*	*	*	*	AIR DATA
					0	1	0	0	INERTIAL
					¢	¢	0	0	AUTOLAND
					0	0	0	0	ACTIVE FLUTTER CONTROL
					0	0	0	0	ENGINE CONTROL
					0	0	0	0	ATTITUDE CONTROL
					¢	¢	1	¢	WEATHER
a(3)	*	*	g	E				Ø	(NULL ARRAY)
a(3)	*	*	h	A				Ø	(NULL ARRAY)
a(3)	*	*	h	B	*	*	*	*	AIDS
					*	*	*	*	VOR/DME
					*	*	*	*	AIR DATA
					0	0	0	*	INERTIAL
					¢	¢	1	*	AUTOLAND
					0	0	0	0	ACTIVE FLUTTER CONTROL
					0	0	0	0	ENGINE CONTROL
					0	0	0	0	ATTITUDE CONTROL
					¢	¢	1	¢	WEATHER
a(3)	*	*	h	C				Ø	(NULL ARRAY)
a(3)	*	*	h	D				Ø	(NULL ARRAY)
a(3)	*	*	h	E				Ø	(NULL ARRAY)
a(3)	*	*	i	A				Ø	(NULL ARRAY)

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

Names are defined in Table 2.

For each row, the resultant level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 1 BASED CAPABILITY FUNCTION

Table 3
Page 19 of 22

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS
	ECONOMICS	OPERATIONS	PROFILE	SAFETY	
a(3)	*	*	i	B	* * * * AIDS * * 0 * VOR/DME * * 0 * AIR DATA 0 0 1 * INERTIAL ¢ ¢ 1 * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 1 ¢ WEATHER
a(3)	*	*	i	C	* * * * AIDS * * 0 * VOR/DME * * 0 * AIR DATA 0 0 1 * INERTIAL ¢ ¢ 0 * AUTOLAND 0 0 0 0 ACTIVE FLUTTER CONTROL 0 0 0 0 ENGINE CONTROL 0 0 0 0 ATTITUDE CONTROL ¢ ¢ 1 ¢ WEATHER
a(3)	*	*	i	D	ø (NULL ARRAY)
a(3)	*	*	i	E	ø (NULL ARRAY)
a(4)	*	*	*	f	* * * * AIDS * * * * VOR/DME * * * * AIR DATA * * * * INERTIAL ¢ ¢ * * AUTOLAND 1 * * * ACTIVE FLUTTER CONTROL * * * * ENGINE CONTROL * * * * ATTITUDE CONTROL ¢ ¢ * ¢ WEATHER
a(4)	*	*	*	g	* * * * AIDS * * * * VOR/DME * * * * AIR DATA * * * * INERTIAL ¢ ¢ * * AUTOLAND 0 * * * ACTIVE FLUTTER CONTROL 1 * * * ENGINE CONTROL * * * * ATTITUDE CONTROL ¢ ¢ * ¢ WEATHER

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

Names are defined in Table 2.

For each row, the resultant Level 1 trajectory set is the intersection of the sets named in Column 2.

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS				
	ECONOMICS	OPERATIONS	PROFILE	SAFETY					
a(4)	*	*	*	h	*	*	*	*	AIDS
					*	*	*	*	VOR/DME
					*	*	*	*	AIR DATA
					*	*	*	*	INERTIAL
					¢	¢	*	*	AUTOLAND
					0	*	*	*	ACTIVE FLUTTER CONTROL
					0	*	*	*	ENGINE CONTROL
					1	*	*	*	ATTITUDE CONTROL
					¢	¢	*	¢	WEATHER
a(4)	*	*	*	i	*	*	*	*	AIDS
					*	*	*	*	VOR/DME
					*	*	*	*	AIR DATA
					0	*	*	*	INERTIAL
					¢	¢	*	*	AUTOLAND
					0	1	*	*	ACTIVE FLUTTER CONTROL
					0	*	*	*	ENGINE CONTROL
					0	*	*	*	ATTITUDE CONTROL
					¢	¢	*	¢	WEATHER
a(4)	*	*	*	j	*	*	*	*	AIDS
					*	*	*	*	VOR/DME
					*	*	*	*	AIR DATA
					0	*	*	*	INERTIAL
					¢	¢	*	*	AUTOLAND
					0	0	*	*	ACTIVE FLUTTER CONTROL
					0	1	*	*	ENGINE CONTROL
					0	*	*	*	ATTITUDE CONTROL
					¢	¢	*	¢	WEATHER
a(4)	*	*	*	k	*	*	*	*	AIDS
					*	*	*	*	VOR/DME
					*	*	*	*	AIR DATA
					0	*	*	*	INERTIAL
					¢	¢	*	*	AUTOLAND
					0	0	*	*	ACTIVE FLUTTER CONTROL
					0	0	*	*	ENGINE CONTROL
					0	1	*	*	ATTITUDE CONTROL
					¢	¢	*	¢	WEATHER

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

Names are defined in Table 2.

For each row, the resultant level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 1 BASED
CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS	
	ECONOMICS	OPERATIONS	PROFILE	SAFETY		
a(4)	*	*	*	l	* * * * * * * * * * * * 0 * * * ¢ ¢ * * 0 0 1 * 0 0 * * 0 0 * * ¢ ¢ * ¢	AIDS VOR/DME AIR DATA INERTIAL AUTOLAND ACTIVE FLUTTER CONTROL ENGINE CONTROL ATTITUDE CONTROL WEATHER
a(4)	*	*	*	m	* * * * * * * * * * * * 0 * * * ¢ ¢ * * 0 0 0 * 0 0 1 * 0 0 * * ¢ ¢ * ¢	AIDS VOR/DME AIR DATA INERTIAL AUTOLAND ACTIVE FLUTTER CONTROL ENGINE CONTROL ATTITUDE CONTROL WEATHER
a(4)	*	*	*	n	* * * * * * * * * * * * 0 * * * ¢ ¢ * * 0 0 0 * 0 0 0 * 0 0 1 * ¢ ¢ * ¢	AIDS VOR/DME AIR DATA INERTIAL AUTOLAND ACTIVE FLUTTER CONTROL ENGINE CONTROL ATTITUDE CONTROL WEATHER
a(4)	*	*	*	o	* * * * * * * * * * * * 0 * * * ¢ ¢ * * 0 0 0 1 0 0 0 * 0 0 0 * ¢ ¢ * ¢	AIDS VOR/DME AIR DATA INERTIAL AUTOLAND ACTIVE FLUTTER CONTROL ENGINE CONTROL ATTITUDE CONTROL WEATHER

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

Names are defined in Table 2.

For each row, the resultant level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 1 BASED
CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 1 PRODUCT TERMS				RESULTING LEVEL 1 TRAJECTORY SETS				
	ECONOMICS	OPERATIONS	PROFILE	SAFETY					
a(4)	*	*	*	p	*	*	*	*	AIDS
					*	*	*	*	VOR/DME
					*	*	*	*	AIR DATA
					0	*	*	*	INERTIAL
					¢	¢	*	*	AUTOLAND
					0	0	0	0	ACTIVE FLUTTER CONTROL
					0	0	0	1	ENGINE CONTROL
					0	0	0	*	ATTITUDE CONTROL
					¢	¢	*	¢	WEATHER
a(4)	*	*	*	q	*	*	*	*	AIDS
					*	*	*	*	VOR/DME
					*	*	*	*	AIR DATA
					0	*	*	*	INERTIAL
					¢	¢	*	*	AUTOLAND
					0	0	0	0	ACTIVE FLUTTER CONTROL
					0	0	0	0	ENGINE CONTROL
					0	0	0	1	ATTITUDE CONTROL
					¢	¢	*	¢	WEATHER
a(4)	*	*	*	r	*	*	*	*	AIDS
					*	*	*	*	VOR/DME
					*	*	*	*	AIR DATA
					0	*	0	*	INERTIAL
					¢	¢	0	1	AUTOLAND
					0	0	0	0	ACTIVE FLUTTER CONTROL
					0	0	0	0	ENGINE CONTROL
					0	0	0	0	ATTITUDE CONTROL
					¢	¢	1	¢	WEATHER
a(4)	*	*	*	s	*	*	*	*	AIDS
					*	*	*	*	VOR/DME
					*	*	*	*	AIR DATA
					0	*	0	1	INERTIAL
					¢	¢	0	0	AUTOLAND
					0	0	0	0	ACTIVE FLUTTER CONTROL
					0	0	0	0	ENGINE CONTROL
					0	0	0	0	ATTITUDE CONTROL
					¢	¢	1	¢	WEATHER

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise B
Column 3: Phase 3 = Cruise C
Column 4: Phase 4 = Landing

Names are defined in Table 2.

For each row, the resultant level 1 trajectory set is the intersection of the sets named in Column 2.

1. ECONOMICS
2. OPERATIONS
3. PROFILE
4. SAFETY.

Using Table 2 to determine the sets $(\xi_i\kappa_1)^{-1}(\xi_i(U_i))$ and designating the Cartesian components of these sets by their Table 2 names,

$$(\xi_i\kappa_1)^{-1}(0) = A \qquad \text{(p. 1, row 1)}$$
$$(\xi_i\kappa_2)^{-1}(0) = A \qquad \text{(p. 2, row 1)}$$
$$(\xi_i\kappa_3)^{-1}(0) = A \cup B \cup C \qquad \text{(p. 4, rows 1-3)}$$
$$(\xi_i\kappa_5)^{-1}(0) = A \cup B \cup C \cup D \cup E$$

(p. 5, last row; p. 6 rows 1-4).

Note that the set names used in Table 2 are "coordinate sensitive," e.g., the A's appearing above mean different trajectory sets for different coordinates. To illustrate the remaining computations, we resolve these ambiguities by adding subscripts, i.e.,

$$(\xi_i\kappa_1)^{-1}(0) = A_1$$
$$(\xi_i\kappa_2)^{-1}(0) = A_2$$
$$(\xi_i\kappa_3)^{-1}(0) = A_3 \cup B_3 \cup C_3$$
$$(\xi_i\kappa_4)^{-1}(0) = A_4 \cup B_4 \cup C_4 \cup D_4 \cup E_4.$$

Accordingly, performing the intersection of equations (3.3.3) (where the   symbols are omitted):

$$\gamma_1^{-1}(a_0) = A_1A_2A(A_3 \cup B_3 \cup C_3)(A_4 \cup B_4 \cup C_4 \cup D_5 \cup E_4)$$
$$= A_1A_2A_3A_4 \cup A_1A_2A_3B_4 \cup \dots \cup A_1A_2C_3E_4.$$

Note that the above expression contains 15 "product terms" when fully written. Note also that, by developing the intersections in the above manner, the subscripts are indeed redundant (i.e., the position of a letter is enough to resolve its meaning). Referring to Table 3, each row of the table beginning with entry a(0) corresponds to a product term in the above expression; the second column of the table displays the corresponding term (with subscripts removed); and the third column gives the resulting intersection of level 1 sets named by letters in the product term. (Since all letters name Cartesian sets and since the intersection of Cartesian sets is Cartesian, the resulting set is Cartesian). $\gamma_1^{-1}(a_0)$, then, is just the union of all the column 3 entries of rows beginning with a(0). Since all but four of these entries are null, the set of all level 1 trajectories corresponding to accomplishment level a_0 is given by

$$\gamma_1^{-1}(a_0) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & * \\ \emptyset & \emptyset & 0 & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \emptyset & \emptyset & 0 & \emptyset \end{bmatrix} \cup \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \emptyset & \emptyset & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \emptyset & \emptyset & 1 & \emptyset \end{bmatrix} \begin{array}{l} \text{AIDS} \\ \text{VOR/DME} \\ \text{AIR DATA} \\ \text{INERTIAL} \\ \text{AUTOLAND} \\ \text{ACTIVE FLUTTER CONTROL} \\ \text{ENGINE CONTROL} \\ \text{ATTITUDE CONTROL} \\ \text{WEATHER} \end{array}$$

$$\cup \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & * \\ \emptyset & \emptyset & * & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \emptyset & \emptyset & 0 & \emptyset \end{bmatrix} \cup \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & * \\ \emptyset & \emptyset & 1 & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \emptyset & \emptyset & 0 & \emptyset \end{bmatrix} \begin{array}{l} \text{AIDS} \\ \text{VOR/DME} \\ \text{AIR DATA} \\ \text{INERTIAL} \\ \text{AUTOLAND} \\ \text{ACTIVE FLUTTER CONTROL} \\ \text{ENGINE CONTROL} \\ \text{ATTITUDE CONTROL} \\ \text{WEATHER .} \end{array}$$

This concludes the example.

Table 3 is therefore a complete description of how behavior of the level 0 model relates to behavior of the level 1 model. In deriving the algorithm used to compute Table 3, emphasis was placed on finding a practical method that would work as opposed to one that would work most efficiently. The efficiency issue is one which must certainly be addressed at some future time, but for the present, we are primarily concerned with establishing the feasibility of the methodology.

The next section describes the computer model which comprises the bottom level of the model hierarchy.

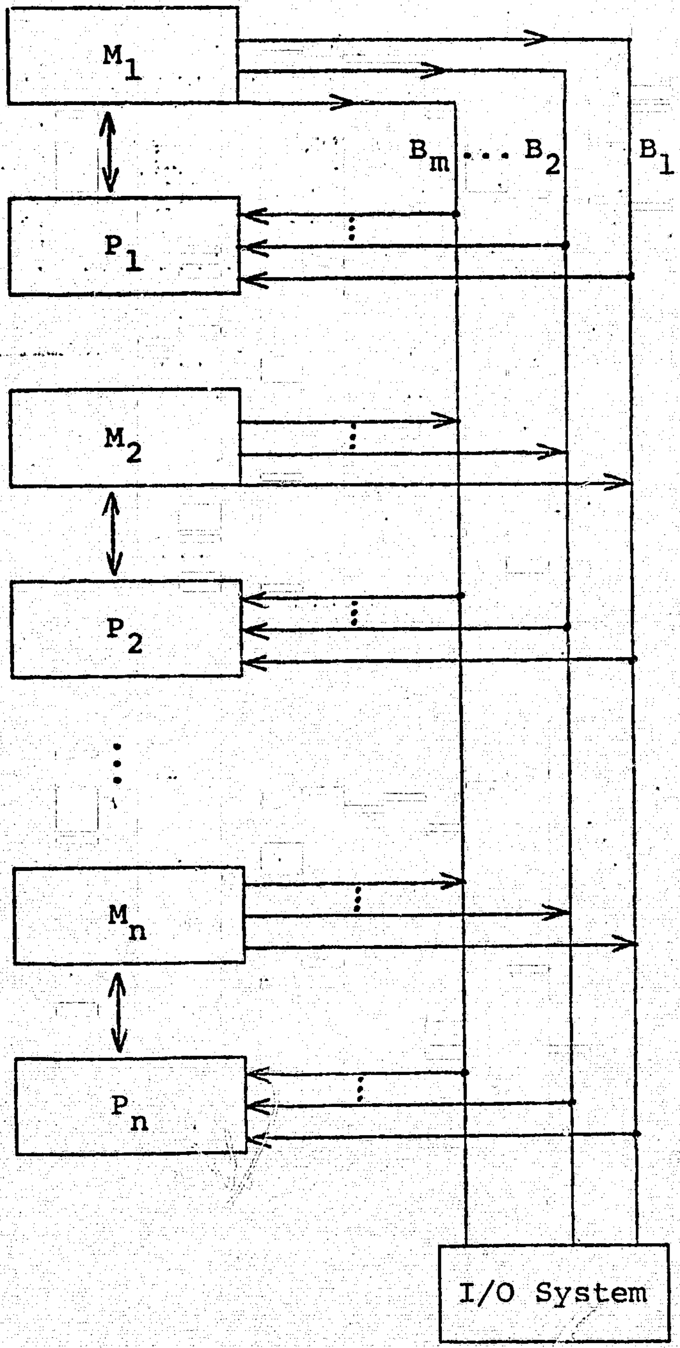
3.3.2 Computer Level Model

A computer model of the SIFT system is described in this section. The model employed is a non-homogeneous Markov process where such processes have been discussed in previous reports (see [2], Section 3.4.2 and [3], Section 3.4.3). The purpose of the computer model is to provide a description of the probabilistic nature of the SIFT system, which is able to change its configuration due to phase changes or to failures occurring during use. As compared with the Markov model for the SIFT system described in [8], the salient feature of the model is that the partitioning of the system is based on both the system's available resources as well as the computational requirements of a given phase. Moreover, since the SIFT system reconfigures in accordance with the task allocation algorithm, we believe that the model should also be tailored to the specific task allocation algorithm selected. Accordingly, the Markov model described here has the advantages that (i) it is more compatible with the higher level models developed in Section 3.3.1 of this report, (ii) the level of detail of the model depends on the user's application.

In the following discussion, it is assumed, as in [8], that the SIFT computer is comprised of a number of processor-memory modules connected to the busses as shown in Figure 4. It is also assumed that the detection and location of processor and bus failures is carried out by the method described in Chapter IV of [8]. In order to relate the state behavior of the bottom model (level 2) to that of the aircraft functional task model (level 1) the phases of the level 1 model are further decomposed into eight phases at level 2, as shown in Table 4.

FIGURE 4
SIFT Configuration

M_i: memory P_i: processor B_i: bus



Level 2 Phases		Corresponding Level 1 Phases	
Phase No.	Description	Phase No.	Description
1	Take-off	1	Take-off/Cruise A
2	Climb		
3	Cruise I		
4	Cruise II	2	Cruise B
5	Cruise III	3	Cruise C
6	Decent		
7	Approach		
8	Landing	4	Landing

TABLE 4

Level 1 and Level 2 Phases

3.3.2.1 Task Allocation

In order to derive the computer model for the SIFT system, it is necessary that the state space of the model be refined enough to distinguish different levels of degraded performance for the system. This condition can be satisfied when the states of the model are chosen to represent different processing capabilities of the system. For example, the state "5 processor and 4 busses" insures that all tasks required to support take-off phase can be accomplished by the system. On the other hand, the state "4 processors and 4 busses" accomodates a reduced workload wherein Inertial System computations are discarded. Since the task profiles are different for different phases, relations between different states and system processing abilities must be established for each phase.

System reconfiguration can occur in the SIFT computer due to phase change, pilot intervention, fault detection and location, etc. It is determined by the Local Executive and the Global Executive using a precomputed task allocation algorithm. Although the feasibility of such a task allocation algorithm has been demonstrated in [8], it is not completely specified. For the purpose of developing a computer model, we have applied the basic design principles described for the SIFT system to develop a workable task allocation algorithm. This allocation algorithm is then accounted for (see Section 3.3.2.2) in the derivation of the Markov process representation of the computer.

The basic parameters for determining a task allocation algorithm are the size of the processor-memory modules and the computational and reliability requirements for each task. Since the set of flight-related application tasks used in the derivation of the above higher level models is a subset of the task set considered in [8], the size of the processor and memory units is scaled down proportionally to account for the reduced workload. It will be assumed in the following discussion that each processor-memory unit has 0.16 MIPS (millions of instructions per second) capacity and has a 5 kiloword memory. However, the computational and reliability requirements for each task are taken directly from [8]. A summary of the tasks considered and their requirements is given in Table 5.

The criticality levels described in Table 5 indicate the degree of reliability required for each task. It can be interpreted as follows (see [7], page 5):

- Criticality Level 1- A function immediately critical to the safty of the flight.
- Criticality Level 2- A function that will be critical to the safty of the flight at some future time during the mission.
- Criticality Level 3- A function whose loss requires a significant change in mission to avoid degradation of safty.
- Criticality Level 4- A function whose loss imposes substantial operational penalties on air crew or ATC.
- Criticality Level 5- A function whose loss has undesirable economic consequences but no significant safty degradation or operational penalty.

The notion of criticality level is used in the SIFT design to assure even distribution of tasks and orderly degradation in reconfiguration. A less critical task may be abandoned when the amount of processor and memory resources have decreased as

TABLE 5
Task Module Properties for Task Allocation

TASK	MIPS	MEMORY (words)	CRITICALITY LEVEL
AFC (Active Flutter Control)	0.069	92	1
AC (Altitude Control)	0.023	2075	1
AUT (Autoland)	0.055	1025	1
EC (Engin Control)	0.119	1500	1
IN (Inertial System)	0.034	2250	3
VOR (VOR/DME Radio)	0.004	300	4
AD (Air Data)	0.001	135	4
AIDS (Aircraft Integrated Data System)	0.002	1300	5
LE (Local Executive)	0.034	320	1
GE (Global Executive)	0.001	1100	1

the result of hardware failures. A highly critical task must either be reassigned to another processor-memory module, or adequate backup systems must be activated. However, the criticality level of a task also depends on the task profile of each phase. For example, although the autoland function has criticality level 1, it is not needed during the take-off phase. Hence, autoland is not allocated during that phase. The task profiles of the flight phases that can influence the criticality level are tabulated in Table 6.

For each phase, it will be assumed that a primary task has a higher priority than a secondary task which, in turn, has a higher priority than a backup task. Accordingly, Table 6 can now be combined with the criticality levels to establish the priority of each task in the task allocation algorithm. When the system must function with a degraded performance, lower priority tasks are discarded first. To obtain the priority ordering, tasks are first ordered in accordance with Table 6, and then ordered according to criticality levels. The resulting priority ordering for each phase is summarized in Table 7.

A task allocation algorithm can now be defined using the method suggested in [8]. Although the method may not be optimal in the sense that it may not yield the highest performability, it achieves some degree of balance in workload distribution. A flowchart representing this method is given in Figure 5. The flowchart is fully explained in [8] except for the reallocation procedure which includes the following steps.

TABLE 6
Task Profiles

Task	Takeoff	Climb Descent	Cruise	Initial Approach	Landing
AFC (Active Flutter Control)	-	P	P	-	-
AC (Altitude Control)	-	P	P	P	-
AUT (Autoland)	-	-	-	-	P
EC (Engin Control)	P	P	P	P	P
IN (Inertial System)	P	P	P	P	P
VOR (VOR/DME Radio)	-	P	P	P	P
AD (Air Data)	B	B	B	B	B
AIDS (Aircraft Integrated Data System)	S	S	S	S	S

Key

P: Prime
S: Secondary
B: Backup
-: Not Applicable

C-2

TABLE 7
Priority Ordering of Tasks

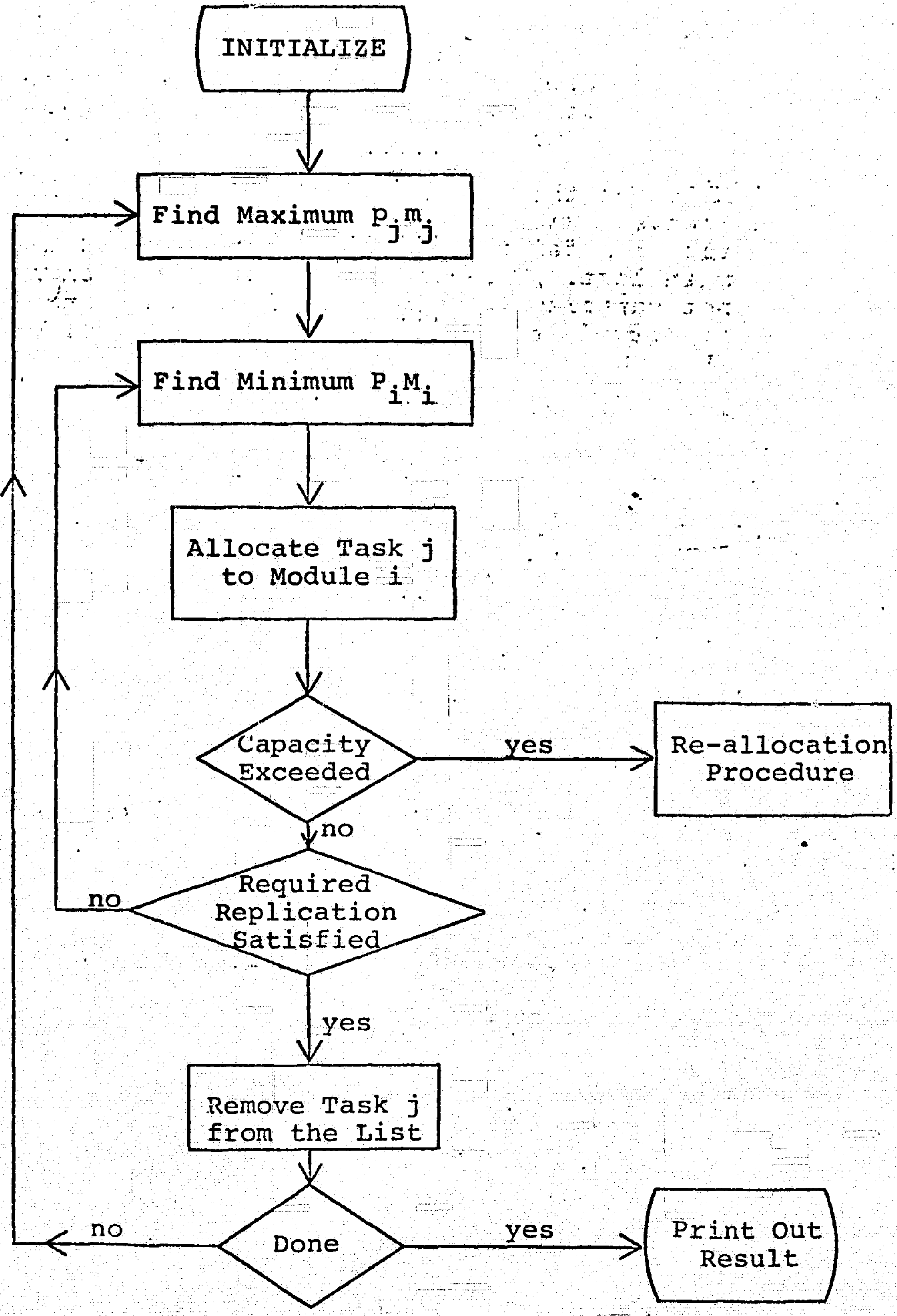
Takeoff: EC > IN > AIDS > AD

Climb, Descent or Cruise: AFC = AC = EC > IN > VOR > AIDS > AD

Initial Approach: AC = EC > IN > VOR > AIDS > AD

Landing: AUT = EC > IN > VOR > AIDS > AD

FIGURE 5
Task Allocation Algorithm



Key:
p_j: Processor Requirement for Task j
m_j: Memory Requirement for Task j
P_i: Processing Load Already Allocated to Module i
M_i: Memory Load Already Allocated to Module i

- Allocate Local Executive to every module, and allocate Global Executive to three modules.
- Allocate remaining tasks triplicated in accordance with Table 7 and beginning with high priority tasks.
- Duplication of tasks are permitted, when resources are exhausted. However, each module has to contain at least one triplicated task (other than GE or LE) to facilitate fault identification.
- Allocate the remaining tasks without replication. However, when a processor containing a non-replicated task has failed, the task is considered to have been lost. This is because simplex assignment is not capable of fault-detection. Thus, the number of missed iterations may be intolerable.

Applying the above algorithm to the cruise phase, the results of the allocation, based on the number of available processors, are given in Tables 8 - 12. In Table 8, all tasks are allocated. Hence, knowing that there are 6 processors and at least 2 busses available, it suffices to infer that all tasks are operational. When 5 processors and at least 2 busses are available, it can be determined that the AIDS system has failed (see Table 9). Similarly, "4 processors and at least 2 busses" and "3 processors and at least 2 busses" can be associated with failure of the Inertial and AIDS systems (see Tables 10-11).

When two processors and at least two busses are available, there may exist two situations. Note that in Table 11 Engin Control is not replicated. Hence, when processor 1 has failed before processor 2 and processor 3, the Engin Control may be erroneously computed, resulting in an excessive number of missed iterations. Consequently, assuming that the failure is correctly detected with some time delay, the above situation can be inter-

TABLE 8
Distributed Assignment of Cruise Phase Tasks Over Six Processor-Memory Units

	Accumulated Task MIPS Per Processor						Accumulated Task Memory Per Memory Unit					
	1	2	3	4	5	6	1	2	3	4	5	6
LE (Local Executive)	.034	.034	.034	.034	.034	.034	320	320	320	320	320	320
GE (Global Executive)	.035	.035	.035	-	-	-	1420	1420	1420	-	-	-
EC (Engin Control)	-	-	-	.153	.153	-	-	-	-	1820	1820	-
AFC (Active Flutter Control)	.104	.104	-	-	-	.103	1512	1512	-	-	-	412
AC (Altitude Control)	.127	.127	-	-	-	.126	3587	3587	-	-	-	2487
IN (Inertial System)	-	-	.069	-	-	.160	-	-	3670	-	-	4737
VOR (VOR/DME)	.131	-	-	.157	.157	-	3887	-	-	2120	2120	-
AD (Air Data)	-	.128	-	.158	.158	-	-	3722	-	2255	2255	-
AIDS (Aircraft Integrated Data System)	-	-	.071	.160	.160	-	-	-	4970	3555	3555	-

TABLE 9
Distributed Assignment of Cruise Phase Tasks Over Five Processor-Memory Units

Task	Accumulated Task MIPS Per Processor					Accumulated Task Memory Per Memory Unit				
	1	2	3	4	5	1	2	3	4	5
LE (Local Executive)	0.034	0.034	0.034	0.034	0.034	320	320	320	320	320
GE (Global Executive)	0.035	0.035	0.035	-	-	1420	1420	1420	-	-
EC (Engin Control)	0.154	-	-	0.153	-	2920	-	-	1820	-
AFC (Active Flutter Control)	-	0.104	0.104	-	0.103	-	1512	1512	-	412
AC (Altitude Control)	-	0.127	-	-	0.126	-	3587	-	-	2487
IN (Inertial System)	-	-	0.138	-	0.160	-	-	3762	-	4737
VOR (VOR/DME)	0.158	0.131	-	0.157	-	3220	3887	-	2120	-
AD (Air Data)	0.159	-	0.139	0.158	-	3355	-	3897	2255	-
AIDS (Aircraft Integrated Data System)	discarded					discarded				

TABLE 10
Distributed Assignment of Cruise Phase Tasks Over Four Processor-Memory Units

Task	Accumulated Task MIPS Per Processor				Accumulated Task Memory Per Memory Unit			
	1	2	3	4	1	2	3	4
LE (Local Executive)	0.034	0.034	0.034	0.034	320	320	320	320
GE (Global Executive)	0.035	0.035	0.034	-	1420	1420	1420	-
EC (Engin Control)	0.154	-	-	0.153	2920	-	-	1820
AFC (Active Flutter Control)	-	0.104	0.104	-	-	1512	1512	-
AC (Altitude Control)	-	0.127	0.127	-	-	3587	3587	-
VOR (VOR/DME)	0.158	0.131	-	0.157	3220	3887	-	2120
AD (Air Data)	0.159	-	0.128	0.158	3355	-	3722	2255
IN (Inertial System)	discarded				discarded			
AIDS (Aircraft Integrated Data System)	discarded				discarded			

TABLE 11
Distributed Assignment of Cruise Phase Tasks Over Three Processor-Memory Units

	Accumulated Task MIPS Per Processor			Accumulated Task Memory Per Memory Unit		
Task	1	2	3	1	2	3
LE (Local Executive)	0.034	0.034	0.034	320	320	320
GE (Global Executive)	0.035	0.035	0.035	1420	1420	1420
EC (Engin Control)	0.154	-	-	2920	-	-
AFC (Active Flutter Control)	-	0.104	0.104	-	1512	1512
AC (Altitude Control)	-	0.127	0.127	-	3587	3587
VOR (VOR/DME)	0.158	0.131	0.131	3220	3887	3887
AD (Air Data)	0.159	0.132	0.132	3355	4022	4022
IN (Inertial System)	discarded			discarded		
AIDS (Aircraft Integrated Data System)	discarded			discarded		

TABLE 12
Distributed Assignment of Cruise Phase Tasks Over Two Processor-Memory Units

Task	Accumulated Task MIPS Per Processor		Accumulated Task Memory Per Memory Unit	
	1	2	1	2
LE (Local Executive)	0.034	0.034	320	320
GE (Global Executive)	0.035	0.035	1420	1420
EC (Engin Control)	0.154	-	2920	-
AFC (Active Flutter Control)	-	0.104	-	1512
AC (Altitude Control)	-	0.127	-	3587
VOR (VOR/DME)	0.158	0.131	3220	3887
[illegible] Integrated [illegible] System)	discarded		discarded	

preted as failure of the Engin Control, together with other tasks discarded in Tables 11-12. However, when processor 2 or processor 3 has failed before processor 1, the failure will be detected immediately and reconfiguration will be initiated. In this case, only the Inertial and AIDS tasks would have been lost, but not Engin Control. Accordingly, these two situations must be distinguished in the computer model (see Section 3.3.2.2). Applying the allocation algorithm to all phases, tasks lost through reconfiguration are shown in Table 13.

3.3.2.2 The Phased Computer Model

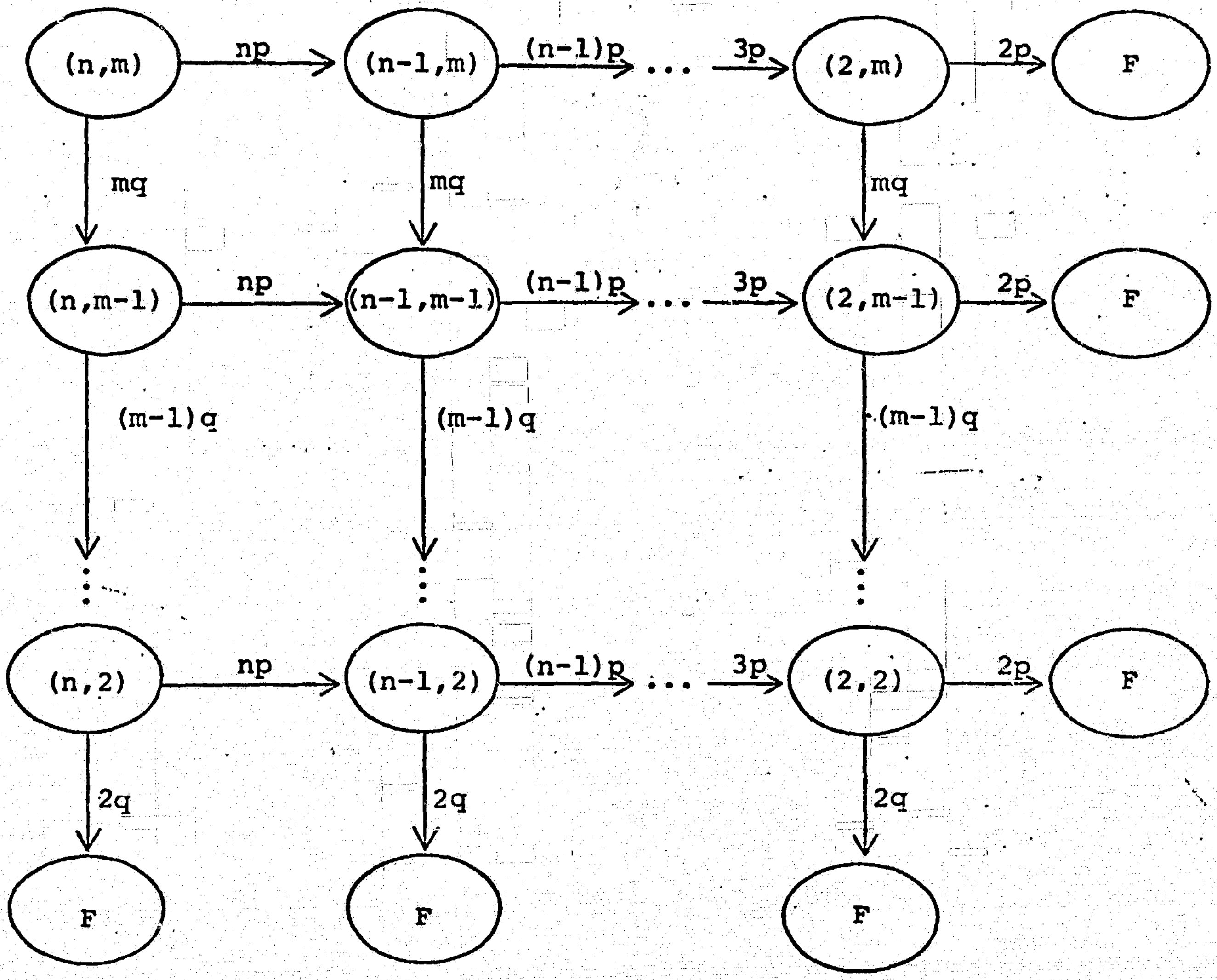
As indicated in the previous section, the probabilistic nature of the computer is represented by a non-homogeneous Markov process. As discussed, the state space of this Markov process is selected in accordance with (i) compliance of the task allocation algorithm with the higher level models and (ii) preservation of the Markov properties via the failure characteristics of the hardware components.

During the takeoff phase (phase 1), the computer is represented by a Markov model with a state transition graph as illustrated in Figure 6. Each state of the graph (except F) represents a specific number of fault-free resources. More precisely, state (i,j) represents a configuration consisting of i fault-free processors and j fault-free busses. State F represents any other configuration. Using Table 13, the state q of the computer at the end of phase 1 relates to the accomplishment of functional tasks during phase 1 as follows.

TABLE 13
Tasks Lost Through Reconfiguration

No. of Processors	Phase 1	2 - 6	7	8
n	-	-	-	-
⋮	⋮	⋮	⋮	⋮
6	-	-	-	-
5	-	AIDS	-	-
4	Inertial	Inertial and AIDS	AIDS	AIDS
3	Inertial	Inertial and AIDS	AIDS	Air Data and AIDS
2	Inertial	{Engin Control Inertial AIDS} or {Inertial AIDS}	{Engin Control Inertial AIDS} or Inertial	{Engin Control Air Data AIDS} or AIDS
1	All Tasks	All Tasks	All Tasks	All Tasks

FIGURE 6
Markov Transition Graph for Takeoff Phase



Key: p = failure rate for each processor unit

q = failure rate for each bus unit

$$\text{If } q = (i,j) \text{ then } \begin{cases} \text{no degradation if} \\ n \geq i \geq 5,\ m \geq j \geq 2, \\[1ex] \text{Inertial System loss if} \\ 4 \geq i \geq 2,\ m \geq j \geq 2, \end{cases}$$

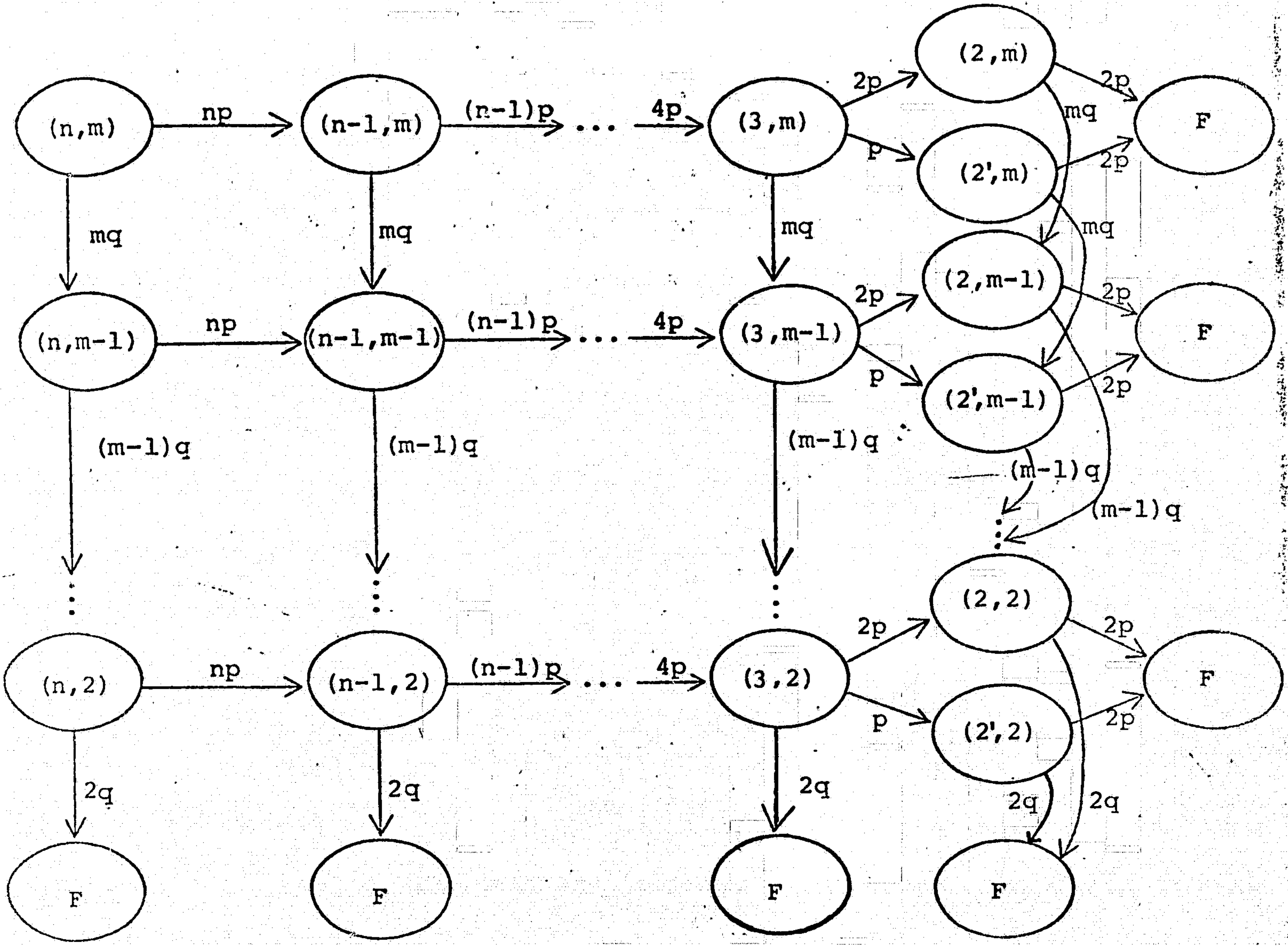
If $q = F$ then all functional tasks lost.

During each of the remaining phases, the computer model is the Markov process with state transition graph shown in Figure 7. Although the underlying Markov processes are the same for these phases, a given state has different effects on level 1 behavior during different phases. For phases 2 - 6 (climb, cruises I-III and descent), the state q at the end of the phase relates to functional task accomplishment as follows:

$$\text{If } q = (i,j) \text{ then } \begin{cases} \text{no degradation if} \\ n \geq i \geq 6,\ m \geq j \geq 2, \\[1ex] \text{AIDS loss if} \\ i = 5,\ m \geq j \geq 2, \\[1ex] \text{Inertial and AIDS loss if} \\ i = 3 \text{ or } 4,\ m \geq j \geq 2, \\[1ex] \text{Engin Control, Inertial and AIDS loss if} \\ i = 2,\ m \geq j \geq 2, \\[1ex] \text{Inertial and AIDS loss if} \\ i = 2,\ m \geq j \geq 2. \end{cases}$$

If $q = F$ then all functional tasks lost.

FIGURE 7
Markov Transition Graph for Phases Other Than Takeoff



Key: p = processor failure rate

q = bus failure rate

For the approach phase (phase 7) the state q at the end of phase 7 implies the following:

$$\text{If } q = (i,j) \text{ then } \begin{cases} \text{no degradation if} \\ n \geq i \geq 5,\ m \geq j \geq 2, \\ \text{AIDS loss if} \\ i = 3 \text{ or } 4,\ m \geq j \geq 2, \\ \text{Engin Control, Inertial and AIDS loss if} \\ i = 2',\ m \geq j \geq 2, \\ \text{Inertial system loss if} \\ i = 2,\ m \geq j \geq 2, \end{cases}$$

If q = F then all functional tasks lost.

Finally, for the landing phase (phase 8) the states have the following implications:

$$\text{If } q = (i,j) \text{ then } \begin{cases} \text{no degradation if} \\ n \geq i \geq 5,\ m \geq j \geq 2, \\ \text{AIDS loss if} \\ i = 4,\ m \geq j \geq 2, \\ \text{Engin Control, Air Data and AIDS loss if} \\ i = 2',\ m \geq j \geq 2, \\ \text{Air data and AIDS loss if} \\ i = 3,\ m \geq j \geq 2, \\ \text{AIDS loss if} \\ i = 2,\ m \geq j \geq 2, \end{cases}$$

If q = F then all functional tasks lost.

Given the implications of level 2 states on level 1 behavior described above, the inverse κ_2^{-1} of the level 2 to level 1 translation κ_2 can be specified. For this purpose, the states of the computer model can be partitioned into seven equivalence classes

$$\{1,2,2',3,4,5,6\}$$

defined as follows:

$$\begin{aligned} 1 &= \{F\} \\ i &= \{(i,j) \mid j \geq 2\} \text{ if } i = 2,2',3,4,5, \\ 6 &= \{(i,j) \mid i \geq 6,\ j \geq 2\}. \end{aligned}$$

The above classification of states is possible because the computational capacity of the system depends only on the number of active processors, as long as at least 2 busses are available.

Table 14 presents κ_2^{-1} in a format similar to Table 2 (κ_1^{-1}), that is, κ_2^{-1} is expressed in terms of its component inverses $((\xi_{ij}\kappa_2)^{-1})$ where i is the task index $(i \leq i \leq 8)$ and j is the number of the level 1 phase $(1 \leq j \leq 4)$. Column 1 gives the coordinate (i,j) under consideration, while column 2 gives the value of the coordinate. The following abbreviations are used to denote the level 1 tasks:

AS = AIDS
VO = VOR/DME
AD = AIR DATA
IN = INERTIAL
AL = AUTOLAND
AF = ACTIVE FLUTTER CONTROL
EC = ENGINE CONTROL
AC = ATTITUDE CONTROL.

COORDINATE INVERSES OF LEVEL 2 TO LEVEL 1 TRANSLATION

LEVEL 1 COORDINATE	VAL		LEVEL 2 TRAJECTORY SETS									NAME
			TAKEOFF	CLIMB	CRUISE I	CRUISE II	CRUISE III	DESCENT	APPROACH	LANDING		
AS(1)	0	[	6	6	6	*	*	*	*	*	]	A
AS(1)	1	[	{1,2,3,4,5}	*	*	*	*	*	*	*	]	b
AS(1)	1	[	6	{1,2,2',3,4,5}	*	*	*	*	*	*	]	c
AS(1)	1	[	6	6	{1,2,2',3,4,5}	*	*	*	*	*	]	d
AS(2)	0	[	*	*	6	6	*	*	*	*	]	A
AS(2)	1	[	*	*	6	{1,2,2',3,4,5}	*	*	*	*	]	b
AS(2)	1	[	*	*	{1,2,2',3,4,5}	*	*	*	*	*	]	c
AS(3)	0	[	*	*	*	6	6	6	{5,6}	*	]	A
AS(3)	1	[	*	*	*	6	{1,2,2',3,4,5}	*	*	*	]	b
AS(3)	1	[	*	*	*	{1,2,2',3,4,5}	*	*	*	*	]	c
AS(3)	1	[	*	*	*	6	6	{1,2,2',3,4,5}	*	*	]	d
AS(3)	1	[	*	*	*	6	6	6	{1,2,2',3,4}	*	]	e
AS(4)	0	[	*	*	*	*	*	*	{5,6}	{5,6}	]	A
AS(4)	1	[	*	*	*	*	*	*	{5,6}	{1,2,2',3,4}	]	b
AS(4)	1	[	*	*	*	*	*	*	{1,2,2',3,4}	*	]	c
VO(1)	0	[	{2,3,4,5,6}	{2,2',3,4,5,6}	{2,2',3,4,5,6}	*	*	*	*	*	]	A
VO(1)	1	[	1	*	*	*	*	*	*	*	]	b
VO(1)	1	[	{2,3,4,5,6}	1	*	*	*	*	*	*	]	c
VO(1)	1	[	{2,3,4,5,6}	{2,2',3,4,5,6}	1	*	*	*	*	*	]	d
VO(2)	0	[	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	*	*	*	*	]	A
VO(2)	1	[	*	*	*	1	*	*	*	*	]	b
VO(2)	1	[	*	*	1	{2,2',3,4,5,6}	*	*	*	*	]	c
VO(3)	0	[	*	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	{2,2',3,4,5,6}	{2,2',3,4,5,6}	*	]	A
VO(3)	1	[	*	*	*	*	1	*	*	*	]	b
VO(3)	1	[	*	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	1	*	*	]	c
VO(3)	1	[	*	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	{2,2',3,4,5,6}	1	*	]	d
VO(3)	1	[	*	*	*	1	{2,2',3,4,5,6}	*	*	*	]	e
VO(4)	0	[	*	*	*	*	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	]	A
VO(4)	1	[	*	*	*	*	*	*	*	1	]	b
VO(4)	1	[	*	*	*	*	*	*	1	{2,2',3,4,5,6}	]	c
AD(1)	0	[	{2,3,4,5,6}	{2,2',3,4,5,6}	{2,2',3,4,5,6}	*	*	*	*	*	]	A
AD(1)	1	[	1	*	*	*	*	*	*	*	]	b
AD(1)	1	[	{2,3,4,5,6}	1	*	*	*	*	*	*	]	c
AD(1)	1	[	{2,3,4,5,6}	{2,2',3,4,5,6}	1	*	*	*	*	*	]	d
AD(2)	0	[	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	*	*	*	*	]	A
AD(2)	1	[	*	*	*	1	*	*	*	*	]	b
AD(2)	1	[	*	*	1	{2,2',3,4,5,6}	*	*	*	*	]	c
			TAKEOFF	CLIMB	CRUISE I	CRUISE II	CRUISE III	DESCENT	APPROACH	LANDING		

Due to space considerations, the following abbreviations are used:

ABBREVIATIONS:

AS = AIDS   VO = VOR/DME   AD = AIR DATA   IN = INERTIAL
AL = AUTOLAND   AF = ACTIVE FLUTTER CONTROL
EC = ENGINE CONTROL   AC = ATTITUDE CONTROL

COORDINATE INVERSES OF LEVEL 2 TO LEVEL 1 TRANSLATION

LEVEL 1 COORDINATE	VAL		LEVEL 2 TRAJECTORY SETS									NAME
			TAKEOFF	CLIMB	CRUISE I	CRUISE II	CRUISE III	DESCENT	APPROACH	LANDING		
AD(3)	0	[	*	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	{2,2',3,4,5,6}	{2,2',3,4,5,6}	*	]	A
AD(3)	1	[	*	*	*	*	1	*	*	*	]	b
AD(3)	1	[	*	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	1	*	*	]	c
AD(3)	1	[	*	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	{2,2',3,4,5,6}	1	*	]	d
AD(3)	1	[	*	*	*	1	{2,2',3,4,5,6}	*	*	*	]	e
AD(4)	0	[	*	*	*	*	*	*	{4,5,6}	{4,5,6}	]	A
AD(4)	0	[	*	*	*	*	*	*	{2,2'}	2'	]	B
AD(4)	1	[	*	*	*	*	*	*	{4,5,6}	{1,2,2',3}	]	c
AD(4)	1	[	*	*	*	*	*	*	{1,3}	*	]	d
AD(4)	1	[	*	*	*	*	*	*	{2,2'}	{1,2,3,4,5,6}	]	e
IN(1)	0	[	{5,6}	{5,6}	{5,6}	*	*	*	*	*	]	A
IN(1)	1	[	{1,2,3,4}	*	*	*	*	*	*	*	]	b
IN(1)	1	[	{5,6}	{1,2,2',3,4}	*	*	*	*	*	*	]	c
IN(1)	1	[	{5,6}	{5,6}	{1,2,2',3,4}	*	*	*	*	*	]	d
IN(2)	0	[	*	*	{5,6}	{5,6}	*	*	*	*	]	A
IN(2)	1	[	*	*	*	{1,2,2',3,4}	*	*	*	*	]	b
IN(2)	1	[	*	*	{1,2,2',3,4}	{5,6}	*	*	*	*	]	c
IN(3)	0	[	*	*	*	{5,6}	{5,6}	{5,6}	{3,4,5,6}	*	]	A
IN(3)	1	[	*	*	*	*	{1,2,2',3,4}	*	*	*	]	b
IN(3)	1	[	*	*	*	{1,2,2',3,4}	{5,6}	*	*	*	]	c
IN(3)	1	[	*	*	*	{5,6}	{5,6}	{1,2,2',3,4}	*	*	]	d
IN(3)	1	[	*	*	*	{5,6}	{5,6}	{5,6}	{1,2,2'}	*	]	e
IN(4)	0	[	*	*	*	*	*	*	{3,4,5,6}	{2,2',3,4,5,6}	]	A
IN(4)	1	[	*	*	*	*	*	*	*	1	]	b
IN(4)	1	[	*	*	*	*	*	*	{1,2,2'}	{2,2',3,4,5,6}	]	c
AL(1)	0	[	{2,3,4,5,6}	{2,2',3,4,5,6}	{2,2',3,4,5,6}	*	*	*	*	*	]	A
AL(1)	1	[	1	*	*	*	*	*	*	*	]	b
AL(1)	1	[	{2,3,4,5,6}	1	*	*	*	*	*	*	]	c
AL(1)	1	[	{2,3,4,5,6}	{2,2',3,4,5,6}	1	*	*	*	*	*	]	d
AL(2)	0	[	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	*	*	*	*	]	A
AL(2)	1	[	*	*	*	1	*	*	*	*	]	b
AL(2)	1	[	*	*	1	{2,2',3,4,5,6}	*	*	*	*	]	c
AL(3)	0	[	*	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	{2,2',3,4,5,6}	{2,2',3,4,5,6}	*	]	A
AL(3)	1	[	*	*	*	*	1	*	*	*	]	b
AL(3)	1	[	*	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	1	*	*	]	c
AL(3)	1	[	*	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	{2,2',3,4,5,6}	1	*	]	d
AL(3)	1	[	*	*	*	1	{2,2',3,4,5,6}	*	*	*	]	e
			TAKEOFF	CLIMB	CRUISE I	CRUISE II	CRUISE III	DESCENT	APPROACH	LANDING		

Due to space considerations, the following abbreviations are used:

ABBREVIATIONS:

AS = AIDS
VO = VOR/DME
AD = AIR DATA
IN = INERTIAL
AL = AUTOLAND
AF = ACTIVE FLUTTER CONTROL
EC = ENGINE CONTROL
AC = ATTITUDE CONTROL

ORIGINAL PAGE IS
OF POOR QUALITY

COORDINATE INVERSES OF
LEVEL 2 TO LEVEL 1 TRANSLATION

Table 14
Page 3 of 4

LEVEL 1 CO-ORDINATE	VAL		LEVEL 2 TRAJECTORY SETS									NAME
			TAKEOFF	CLIMB	CRUISE I	CRUISE II	CRUISE III	DESCENT	APPROACH	LANDING		
AL(4)	0	[	*	*	*	*	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	]	A
AL(4)	1	[	*	*	*	*	*	*	*	1	]	b
AL(4)	1	[	*	*	*	*	*	*	1	{2,2',3,4,5,6}	]	c
AF(1)	0	[	{2,3,4,5,6}	{2,2',3,4,5,6}	{2,2',3,4,5,6}	*	*	*	*	*	]	A
AF(1)	1	[	1	*	*	*	*	*	*	*	]	b
AF(1)	1	[	{2,3,4,5,6}	1	*	*	*	*	*	*	]	c
AF(1)	1	[	{2,3,4,5,6}	{2,2',3,4,5,6}	1	*	*	*	*	*	]	d
AF(2)	0	[	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	*	*	*	*	]	A
AF(2)	1	[	*	*	*	1	*	*	*	*	]	b
AF(2)	1	[	*	*	1	{2,2',3,4,5,6}	*	*	*	*	]	c
AF(3)	0	[	*	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	{2,2',3,4,5,6}	{2,2',3,4,5,6}	*	]	A
AF(3)	1	[	*	*	*	*	1	*	*	*	]	b
AF(3)	1	[	*	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	1	*	*	]	c
AF(3)	1	[	*	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	{2,2',3,4,5,6}	1	*	]	d
AF(3)	1	[	*	*	*	1	{2,2',3,4,5,6}	*	*	*	]	e
AF(4)	0	[	*	*	*	*	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	]	A
AF(4)	1	[	*	*	*	*	*	*	*	1	]	b
AF(4)	1	[	*	*	*	*	*	*	1	{2,2',3,4,5,6}	]	c
EC(1)	0	[	{2,3,4,5,6}	{2',3,4,5,6}	{2',3,4,5,6}	*	*	*	*	*	]	A
EC(1)	1	[	1	*	*	*	*	*	*	*	]	b
EC(1)	1	[	{2,3,4,5,6}	{1,2}	*	*	*	*	*	*	]	c
EC(1)	1	[	{2,3,4,5,6}	{2',3,4,5,6}	{1,2}	*	*	*	*	*	]	d
EC(2)	0	[	*	*	{2',3,4,5,6}	{2',3,4,5,6}	*	*	*	*	]	A
EC(2)	1	[	*	*	*	{1,2}	*	*	*	*	]	b
EC(2)	1	[	*	*	{1,2}	{2',3,4,5,6}	*	*	*	*	]	c
EC(3)	0	[	*	*	*	{2',3,4,5,6}	{2',3,4,5,6}	{2',3,4,5,6}	{2',3,4,5,6}	*	]	A
EC(3)	1	[	*	*	*	*	{1,2}	*	*	*	]	b
EC(3)	1	[	*	*	*	{2',3,4,5,6}	{2',3,4,5,6}	{1,2}	*	*	]	c
EC(3)	1	[	*	*	*	{2',3,4,5,6}	{2',3,4,5,6}	{2',3,4,5,6}	{1,2}	*	]	d
EC(3)	1	[	*	*	*	{1,2}	{2',3,4,5,6}	*	*	*	]	e
EC(4)	0	[	*	*	*	*	*	*	{2',3,4,5,6}	{2',3,4,5,6}	]	A
EC(4)	1	[	*	*	*	*	*	*	*	{1,2}	]	b
EC(4)	1	[	*	*	*	*	*	*	{1,2}	{2',3,4,5,6}	]	c
AC(1)	0	[	{2,3,4,5,6}	{2,2',3,4,5,6}	{2,2',3,4,5,6}	*	*	*	*	*	]	A
AC(1)	1	[	1	*	*	*	*	*	*	*	]	b
AC(1)	1	[	{2,3,4,5,6}	1	*	*	*	*	*	*	]	c
AC(1)	1	[	{2,3,4,5,6}	{2,2',3,4,5,6}	1	*	*	*	*	*	]	d
			TAKEOFF	CLIMB	CRUISE I	CRUISE II	CRUISE III	DESCENT	APPROACH	LANDING		

Due to space considerations, the following abbreviations are used:

ABBREVIATIONS:
AS = AIDS   VO = VOR/DME   AD = AIR DATA   IN = INERTIAL
AL = AUTOLAND   AF = ACTIVE FLUTTER CONTROL
EC = ENGINE CONTROL   AC = ATTITUDE CONTROL

COORDINATE INVERSES OF LEVEL 2 TO LEVEL 1 TRANSLATION

Table 14
Page 4 of 4

LEVEL 1 CO-ORDI-NATE	VAL		LEVEL 2 TRAJECTORY SETS TAKEOFF	CLIMB	CRUISE I	CRUISE II	CRUISE III	DESCENT	APPROACH	LANDING		NAME
AC(2)	0	L	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	*	*	*	*	]	A
AC(2)	1	L	*	*	*	1	*	*	*	*	]	b
AC(2)	1	L	*	*	1	{2,2',3,4,5,6}	*	*	*	*	]	c
AC(3)	0	L	*	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	{2,2',3,4,5,6}	{2,2',3,4,5,6}	*	]	A
AC(3)	1	L	*	*	*	*	1	*	*	*	]	b
AC(3)	1	L	*	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	1	*	*	]	c
AC(3)	1	L	*	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	{2,2',3,4,5,6}	1	*	]	d
AC(3)	1	L	*	*	*	1	{2,2',3,4,5,6}	*	*	*	]	e
[illegible]	0	L	*	*	*	*	*	*	{2,2',3,4,5,6}	{2,2',3,4,5,6}	]	A
		L	*	*	*	*	*	*	*	1	]	b
		L	*	*	*	*	*	*	1	{2,2',3,4,5,6}	]	c
			TAKEOFF	CLIMB	CRUISE I	CRUISE II	CRUISE III	DESCENT	APPROACH	LANDING		

Due to space considerations, the following abbreviations are used:

ABBREVIATIONS:
AS = AIDS VO = VOR/DME AD = AIR DATA IN = INERTIAL
AL = AUTOLAND AF = ACTIVE FLUTTER CONTROL
EC = ENGINE CONTROL AC = ATTITUDE CONTROL

Columns 3-10 then give a trajectory set that maps into the corresponding level 2 value. The union of all level 2 trajectory sets indicated for a given coordinate ij and value v is the preimage $(\xi_{ij}\kappa_2)^{-1}(v)$. For instance the level 2 trajectory set $(\xi_{53}\ \kappa_2)^{-1}(1)$ is the union of all trajectory sets for which INERTIAL(3) = 1, i.e., the set

[* * * * {1,2,2',3,4} * * *]
U [* * * {1,2,2',3,4} {5,6} * * *]
U [* * * {5,6} {5,6} {1,2,2',3,4} * *]
U [* * * {5,6} {5,6} {5,6} {1,2,2'}].

Finally, the last column of table 14 assigns each trajectory set a one letter name. Again, capital letters denote trajectory sets associated with coordinate values of 0; lower case names are used with trajectory sets affiliated with coordinate values of 1. Sets are then referred to by these names in a later table (Table 15).

Using γ_1^{-1} (Table 3) and κ_2^{-1} (Table 14), the desired base model trajectory sets are determined. This step is described in the section that follows.

3.3.3 Derivation of Base Model Trajectory Sets

Given the inverse γ_1^{-1} of the level 1 based capability function γ_1 (Table 3) and given the inverse κ_2^{-1} of the inter-level translation κ_2 (Table 14), the inverse γ_2^{-1} of the level 2 based capability function is determined in a manner similar to the derivation of γ_1^{-1}. Moreover, since level 2 is the bottom level, $\gamma_2^{-1} = \gamma^{-1}$, the inverse the capability function of the total system. More precisely, if a is an accomplishment level, let $U_1 \times V_1, U_2 \times V_2, \ldots, U_m \times V_m$ denote the Cartesian components of the level 1 trajectory set $\gamma_1^{-1}(a)$ (see Table 3). For a particular component $U_k \times V_k$, U_k denotes the "composite part" of the trajectory set (the first eight rows that describe functional task accomplishment) and V_k denotes the "basic part" (the last row that describes WEATHER behavior). Then, as with equation (3.1.3), it follows that

$$\gamma^{-1}(a) = \gamma_2^{-1}(a) = \bigcup_{k=1}^{m} \kappa_2^{-1}(U_k) \times V_k. \qquad (3.3.1)$$

(Note that (3.3.1) differs from (3.1.1) in that the basic trajectories (WEATHER) must be carried down from level 1 to level 2. Also, when V_k is carried down, additional coordinates are added to match the number of phases of the level 2 model.) Each preimage $\kappa_2^{-1}(U_k)$ is then formulated using equation (3.1.2), where in this case the coordinate indices in C are pairs (i,j); i being the i^{th} functional task and j being the j^{th} phase of the level 1 trajectories. Hence

$$\kappa_2^{-1}(U_k) = \bigcap_{ij=1,1}^{8,4} (\xi_{ij}\ \kappa_2)^{-1}(\xi_{ij}(U_k)). \qquad (3.3.2)$$

The values of the intersected terms on the right are determined using Table 14, such that each term is expressed as a union of Cartesian sets. These unions are then intersected according to equation (3.3.2), in the systematic fashion used earlier at level 1. The result is an expression of $\kappa_2^{-1}(U_k)$ as a union of Cartesian trajectory sets. Finally, applying (3.3.1), $\gamma^{-1}(a)$ is just the union of all the $\kappa_2^{-1}(U_k)$ unions ($k = 1,2,\ldots,m$) with the weather trajectories V_k adjoined to each Cartesian component of $\kappa_2^{-1}(U_k)$. These resulting sets are displayed in Table 1 . To illustrate this computation, consider the following example.

Example

Suppose $a = a_0$. Then from Table 3 (or, alternatively, the example at the end of Section 3.3.1.3), $\gamma_1^{-1}(a_0)$ has four Cartesian componints:

$$U_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & * \\ \text{¢} & \text{¢} & 0 & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \text{¢} & \text{¢} & 0 & \text{¢} \end{bmatrix} \qquad U_2 = \begin{array}{l} \left[\begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \text{¢} & \text{¢} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \text{¢} & \text{¢} & 1 & \text{¢} \end{matrix}\right] \end{array} \begin{array}{l} \text{AIDS} \\ \text{VOR/DME} \\ \text{AIR DATA} \\ \text{INERTIAL} \\ \text{AUTOLAND} \\ \text{ACTIVE FLUTTER CONTROL} \\ \text{ENGINE CONTROL} \\ \text{ATTITUDE CONTROL} \\ \text{WEATHER} \end{array}$$

$$U_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & * \\ \text{¢} & \text{¢} & * & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \text{¢} & \text{¢} & 0 & \text{¢} \end{bmatrix} \qquad U_4 = \begin{array}{l} \left[\begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & * \\ \text{¢} & \text{¢} & 1 & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \text{¢} & \text{¢} & 0 & \text{¢} \end{matrix}\right] \end{array} \begin{array}{l} \text{AIDS} \\ \text{VOR/DME} \\ \text{AIR DATA} \\ \text{INERTIAL} \\ \text{AUTOLAND} \\ \text{ACTIVE FLUTTER CONTROL} \\ \text{ENGINE CONTROL} \\ \text{ATTITUDE CONTROL} \\ \text{WEATHER} \end{array}$$

Consider now the trajectory set U_1. $\xi_{ij}(U_1)$ is the i,j^th entry of U_1, and, except for the values * and ¢ (which as argued

LEVEL 2 BASED
CAPABILITY FUNCTION

TABLE 15
PAGE 1 OF 17

ACCOMPLISHMENT LEVEL	LEVEL 2 TRAJECTORY SET NAMES FOR LEVEL 1 TRAJECTORIES					RESULTING LEVEL 2 TRAJECTORY SETS			
a(0)	A	A	A	A	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	A	AIR DATA	CRUISE I	6		¢
	A	A	A	*	INERTIAL	CRUISE II	6	x	¢
	¢	¢	A	*	AUTOLAND	CRUISE III	6		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	6		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{5,6}		¢
a(0)	A	A	A	A	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	A	AIR DATA	CRUISE I	6		¢
	A	A	A	A	INERTIAL	CRUISE II	6	x	¢
	¢	¢	A	A	AUTOLAND	CRUISE III	6		1
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	6		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{5,6}		¢
a(1)	b	*	*	*	AIDS	TAKEOFF	5		¢
	A	A	A	A	VOR/DME	CLIMB	{5,6}		¢
	A	A	A	A	AIR DATA	CRUISE I	{5,6}		¢
	A	A	A	*	INERTIAL	CRUISE II	{5,6}	x	¢
	¢	¢	A	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		¢
a(1)	c	*	*	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	5		¢
	A	A	A	A	AIR DATA	CRUISE I	{5,6}		¢
	A	A	A	*	INERTIAL	CRUISE II	{5,6}	x	¢
	¢	¢	A	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		¢
d(1)	d	*	*	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	A	AIR DATA	CRUISE I	5		¢
	A	A	A	*	INERTIAL	CRUISE II	{5,6}	x	¢
	¢	¢	A	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		¢

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise b
Column 3: Phase 3 = Cruise c
Column 4: Phase 4 = Landing

Names are defined in Table 14.

For each row, the resulting level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 2 BASED
CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 2 TRAJECTORY SET NAMES FOR LEVEL 1 TRAJECTORIES					RESULTING LEVEL 2 TRAJECTORY SETS			
	b	*	*	*	AIDS	TAKEOFF	5		¢
	A	A	A	A	VOR/DME	CLIMB	{5,6}		¢
	A	A	A	A	AIR DATA	CRUISE I	{5,6}		¢
a(1)	A	A	A	A	INERTIAL	CRUISE II	{5,6}	x	¢
	¢	¢	A	A	AUTOLAND	CRUISE III	{5,6}		1
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		¢
	c	*	*	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	5		¢
	A	A	A	A	AIR DATA	CRUISE I	{5,6}		¢
a(1)	A	A	A	A	INERTIAL	CRUISE II	{5,6}	x	¢
	¢	¢	A	A	AUTOLAND	CRUISE III	{5,6}		1
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		¢
	d	*	*	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	A	AIR DATA	CRUISE I	5		¢
a(1)	A	A	A	A	INERTIAL	CRUISE II	{5,6}	x	¢
	¢	¢	A	A	AUTOLAND	CRUISE III	{5,6}		1
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		¢
	b	*	*	*	AIDS	TAKEOFF	5		¢
	A	A	A	A	VOR/DME	CLIMB	{5,6}		¢
	A	A	A	A	AIR DATA	CRUISE I	{5,6}		¢
a(1)	A	A	b	*	INERTIAL	CRUISE II	{5,6}	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	{2',3,4}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4,5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		¢

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise b
Column 3: Phase 3 = Cruise c
Column 4: Phase 4 = Landing

Names are defined in Table 14.

For each row, the resulting level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 2 BASED
CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 2 TRAJECTORY SET NAMES FOR LEVEL 1 TRAJECTORIES					RESULTING LEVEL 2 TRAJECTORY SETS			
	b	*	*	*	AIDS	TAKEOFF	5		∅
	A	A	A	A	VOR/DME	CLIMB	{5,6}		∅
	A	A	A	B	AIR DATA	CRUISE I	{5,6}		∅
a(1)	A	A	b	*	INERTIAL	CRUISE II	{5,6}	x	∅
	∅	∅	*	*	AUTOLAND	CRUISE III	{2',3,4}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4,5,6}		∅
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		∅
	A	A	A	A	ATTITUDE CONTROL	LANDING	2'		∅
	b	*	*	*	AIDS	TAKEOFF	5		∅
	A	A	A	A	VOR/DME	CLIMB	{5,6}		∅
	A	A	A	A	AIR DATA	CRUISE I	{5,6}		∅
a(1)	A	A	d	*	INERTIAL	CRUISE II	{5,6}	x	∅
	∅	∅	*	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4}		∅
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		∅
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		∅
	b	*	*	*	AIDS	TAKEOFF	5		∅
	A	A	A	A	VOR/DME	CLIMB	{5,6}		∅
	A	A	A	B	AIR DATA	CRUISE I	{5,6}		∅
a(1)	A	A	d	*	INERTIAL	CRUISE II	{5,6}	x	∅
	∅	∅	*	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4}		∅
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		∅
	A	A	A	A	ATTITUDE CONTROL	LANDING	2'		∅
	b	*	*	*	AIDS	TAKEOFF	5		∅
	A	A	A	A	VOR/DME	CLIMB	{5,6}		∅
	A	A	A	B	AIR DATA	CRUISE I	{5,6}		∅
a(1)	A	A	e	*	INERTIAL	CRUISE II	{5,6}	x	∅
	∅	∅	*	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		∅
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		∅
	A	A	A	A	ATTITUDE CONTROL	LANDING	2'		∅
	c	*	*	*	AIDS	TAKEOFF	6		∅
	A	A	A	A	VOR/DME	CLIMB	5		∅
	A	A	A	A	AIR DATA	CRUISE I	{5,6}		∅
a(1)	A	A	b	*	INERTIAL	CRUISE II	{5,6}	x	∅
	∅	∅	*	*	AUTOLAND	CRUISE III	{2',3,4}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4,5,6}		∅
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		∅
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		∅

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise b
Column 3: Phase 3 = Cruise c
Column 4: Phase 4 = Landing

Names are defined in Table 14.

For each row, the resulting level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 2 BASED CAPABILITY FUNCTION

TABLE 15
PAGE 4 OF 17

CCOMPLISHMENT LEVEL	LEVEL 2 TRAJECTORY SET NAMES FOR LEVEL 1 TRAJECTORIES					RESULTING LEVEL 2 TRAJECTORY SETS			
a(1)	c	*	*	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	5		¢
	A	A	A	B	AIR DATA	CRUISE I	{5,6}		¢
	A	A	b	*	INERTIAL	CRUISE II	{5,6}	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	{2',3,4}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4,5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	2'		¢
a(1)	c	*	*	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	5		¢
	A	A	A	A	AIR DATA	CRUISE I	{5,6}		¢
	A	A	d	*	INERTIAL	CRUISE II	{5,6}	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		¢
a(1)	c	*	*	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	5		¢
	A	A	A	B	AIR DATA	CRUISE I	{5,6}		¢
	A	A	d	*	INERTIAL	CRUISE II	{5,6}	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	2'		¢
a(1)	c	*	*	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	5		¢
	A	A	A	B	AIR DATA	CRUISE I	{5,6}		¢
	A	A	e	*	INERTIAL	CRUISE II	{5,6}	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	2'		¢
a(1)	d	*	*	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	A	AIR DATA	CRUISE I	5		¢
	A	A	b	*	INERTIAL	CRUISE II	{5,6}	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	{2',3,4}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4,5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		¢

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise b
Column 3: Phase 3 = Cruise c
Column 4: Phase 4 = Landing

Names are defined in Table 14.

For each row, the resulting level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 2 BASED CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 2 TRAJECTORY SET NAMES FOR LEVEL 1 TRAJECTORIES					RESULTING LEVEL 2 TRAJECTORY SETS			
	d	*	*	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	*	B	AIR DATA	CRUISE I	5		¢
a(1)	A	A	b	*	INERTIAL	CRUISE II	{5,6}	x	¢
	¢	¢	A	*	AUTOLAND	CRUISE III	{2',3,4}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4,5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	2'		¢
	d	*	*	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	A	AIR DATA	CRUISE I	5		¢
a(1)	A	A	d	*	INERTIAL	CRUISE II	{5,6}	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		¢
	d	*	*	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	B	AIR DATA	CRUISE I	5		¢
a(1)	A	A	d	*	INERTIAL	CRUISE II	{5,6}	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	2'		¢
	d	*	*	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	D	AIR DATA	CRUISE I	5		¢
a(1)	A	A	e	*	INERTIAL	CRUISE II	{5,6}	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	2'		¢
	A	b	*	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	A	AIR DATA	CRUISE I	6		¢
a(1)	A	A	A	*	INERTIAL	CRUISE II	5	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		¢

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise b
Column 3: Phase 3 = Cruise c
Column 4: Phase 4 = Landing

Names are defined in Table 14.

For each row, the resulting level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 2 BASED
CAPABILITY FUNCTION

TABLE 15
PAGE 6 OF 17

ACCOMPLISHMENT LEVEL	LEVEL 2 TRAJECTORY SET NAMES FOR LEVEL 1 TRAJECTORIES					RESULTING LEVEL 2 TRAJECTORY SETS			
	A	b	*	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	A	AIR DATA	CRUISE I	6		¢
a(1)	A	A	A	A	INERTIAL	CRUISE II	5	x	¢
	¢	¢	A	A	AUTOLAND	CRUISE III	{5,6}		1
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		¢
	A	b	*	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	A	AIR DATA	CRUISE I	6		¢
a(1)	A	A	b	*	INERTIAL	CRUISE II	5	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	{2',3,4}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4,5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		¢
	A	b	*	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	B	AIR DATA	CRUISE I	6		¢
a(1)	A	A	b	*	INERTIAL	CRUISE II	5	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	{2',3,4}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4,5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	2'		¢
	A	b	*	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	A	AIR DATA	CRUISE I	6		¢
a(1)	A	A	d	*	INERTIAL	CRUISE II	5	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		¢
	A	b	*	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	B	AIR DATA	CRUISE I	6		¢
a(1)	A	A	d	*	INERTIAL	CRUISE II	5	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	2'		¢

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise b
Column 3: Phase 3 = Cruise c
Column 4: Phase 4 = Landing

Names are defined in Table 14.

For each row, the resulting level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 2 BASED
CAPABILITY FUNCTION

CCOMPLISHMENT LEVEL	LEVEL 2 TRAJECTORY SET NAMES FOR LEVEL 1 TRAJECTORIES					RESULTING LEVEL 2 TRAJECTORY SETS			
	A	b	*	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	D	AIR DATA	CRUISE I	6		¢
a(1)	A	A	c	*	INERTIAL	CRUISE II	5	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	2'		¢
	A	A	b	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	A	AIR DATA	CRUISE I	6		¢
a(1)	A	A	A	*	INERTIAL	CRUISE II	6	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	5		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		¢
	A	A	d	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	A	AIR DATA	CRUISE I	6		¢
a(1)	A	A	A	*	INERTIAL	CRUISE II	6	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	6		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	5		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		¢
	A	A	e	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	A	AIR DATA	CRUISE I	6		¢
a(1)	A	A	A	*	INERTIAL	CRUISE II	6	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	6		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	6		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	4		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		¢
	A	A	b	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	A	AIR DATA	CRUISE I	6		¢
a(1)	A	A	A	A	INERTIAL	CRUISE II	6	x	¢
	¢	¢	A	A	AUTOLAND	CRUISE III	5		1
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		¢

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise b
Column 3: Phase 3 = Cruise c
Column 4: Phase 4 = Landing

Names are defined in Table 14.

For each row, the resulting level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 2 BASED CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 2 TRAJECTORY SET NAMES FOR LEVEL 1 TRAJECTORIES					RESULTING LEVEL 2 TRAJECTORY SETS			
	A	A	d	*	AIDS	TAKEOFF	6		∅
	A	A	A	A	VOR/DME	CLIMB	6		∅
	A	A	A	A	AIR DATA	CRUISE I	6		∅
a(1)	A	A	A	A	INERTIAL	CRUISE II	6	x	∅
	∅	∅	A	A	AUTOLAND	CRUISE III	6		1
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	5		∅
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		∅
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		∅
	A	A	e	*	AIDS	TAKEOFF	6		∅
	A	A	A	A	VOR/DME	CLIMB	6		∅
	A	A	A	A	AIR DATA	CRUISE I	6		∅
a(1)	A	A	A	A	INERTIAL	CRUISE II	6	x	∅
	∅	∅	A	A	AUTOLAND	CRUISE III	6		1
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	6		∅
	A	A	A	A	ENGINE CONTROL	APPROACH	4		∅
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		∅
									∅
	A	A	b	*	AIDS	TAKEOFF	6		∅
	A	A	A	A	VOR/DME	CLIMB	6		∅
	A	A	A	A	AIR DATA	CRUISE I	6		∅
a(1)	A	A	b	*	INERTIAL	CRUISE II	6	x	∅
	∅	∅	*	*	AUTOLAND	CRUISE III	{2',3,4}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4,5,6}		∅
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		∅
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		∅
	A	A	b	*	AIDS	TAKEOFF	6		∅
	A	A	A	A	VOR/DME	CLIMB	6		∅
	A	A	A	B	AIR DATA	CRUISE I	6		∅
a(1)	A	A	b	*	INERTIAL	CRUISE II	6	x	∅
	∅	∅	*	*	AUTOLAND	CRUISE III	{2',3,4}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4,5,6}		∅
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		∅
	A	A	A	A	ATTITUDE CONTROL	LANDING	2'		∅
	A	A	b	*	AIDS	TAKEOFF	6		∅
	A	A	A	A	VOR/DME	CLIMB	6		∅
	A	A	A	A	AIR DATA	CRUISE I	6		∅
a(1)	A	A	d	*	INERTIAL	CRUISE II	6	x	∅
	∅	∅	*	*	AUTOLAND	CRUISE III	5		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4}		∅
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		∅
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		∅

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise b
Column 3: Phase 3 = Cruise c
Column 4: Phase 4 = Landing

Names are defined in Table 14.

For each row, the resulting level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 2 BASED
CAPABILITY FUNCTION

TABLE 15
PAGE 9 OF 17

ACCOMPLISHMENT LEVEL	LEVEL 2 TRAJECTORY SET NAMES FOR LEVEL 1 TRAJECTORIES					RESULTING LEVEL 2 TRAJECTORY SETS			
	A	A	b	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	B	AIR DATA	CRUISE I	6		¢
a(1)	A	A	d	*	INERTIAL	CRUISE II	6	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	5		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	2'		¢
	A	A	b	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	B	AIR DATA	CRUISE I	6		¢
a(1)	A	A	c	*	INERTIAL	CRUISE II	6	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	5		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	2'		¢
	A	A	d	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	A	AIR DATA	CRUISE I	6		¢
a(1)	A	A	d	*	INERTIAL	CRUISE II	6	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	6		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{4,5,6}		¢
	A	A	d	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	B	AIR DATA	CRUISE I	6		¢
a(1)	A	A	d	*	INERTIAL	CRUISE II	6	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	6		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	2'		¢
	A	A	d	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	B	AIR DATA	CRUISE I	6		¢
a(1)	A	A	e	*	INERTIAL	CRUISE II	6	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	6		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	5		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	2'		¢

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise b
Column 3: Phase 3 = Cruise c
Column 4: Phase 4 = Landing

Names are defined in Table 14.

For each row, the resulting level 1 trajectory set is the intersection of the sets named in Column 2.

ORIGINAL PAGE IS
OF POOR QUALITY

LEVEL 2 BASED
CAPABILITY FUNCTION

TABLE 15
PAGE 10 OF 17

ACCOMPLISHMENT LEVEL	LEVEL 2 TRAJECTORY SET NAMES FOR LEVEL 1 TRAJECTORIES					RESULTING LEVEL 2 TRAJECTORY SETS			
a(1)	A	A	$\underline{e}$	*	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	B	AIR DATA	CRUISE I	6		¢
	A	A	$\underline{e}$	*	INERTIAL	CRUISE II	6	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	6		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	6		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	2'		¢
a(1)	A	A	A	$\underline{b}$	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	A	AIR DATA	CRUISE I	6		¢
	A	A	A	*	INERTIAL	CRUISE II	6	x	¢
	¢	¢	A	*	AUTOLAND	CRUISE III	6		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	6		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	4		¢
a(1)	A	A	A	$\underline{b}$	AIDS	TAKEOFF	6		¢
	A	A	A	A	VOR/DME	CLIMB	6		¢
	A	A	A	A	AIR DATA	CRUISE I	6		¢
	A	A	A	A	INERTIAL	CRUISE II	6	x	¢
	¢	¢	A	A	AUTOLAND	CRUISE III	6		1
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	6		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	4		¢
a(2)	*	*	*	*	AIDS	TAKEOFF	{5,6}		¢
	A	A	A	A	VOR/DME	CLIMB	{5,6}		¢
	A	A	A	$\underline{c}$	AIR DATA	CRUISE I	{5,6}		¢
	A	A	C	*	INERTIAL	CRUISE II	{5,6}	x	¢
	¢	¢	A	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{2',3}		¢
a(2)	*	*	*	*	AIDS	TAKEOFF	{5,6}		¢
	A	A	A	A	VOR/DME	CLIMB	{5,6}		¢
	A	A	A	$\underline{d}$	AIR DATA	CRUISE I	{5,6}		¢
	A	A	A	*	INERTIAL	CRUISE II	{5,6}	x	¢
	¢	¢	A	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	3		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{2',3,4,5,6}		¢

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise b
Column 3: Phase 3 = Cruise c
Column 4: Phase 4 = Landing

Names are defined in Table 14.

For each row, the resulting level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 2 BASED
CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 2 TRAJECTORY SET NAMES FOR LEVEL 1 TRAJECTORIES					RESULTING LEVEL 2 TRAJECTORY SETS			
a(2)	*	*	*	*	AIDS	TAKEOFF	{5,6}	x	¢
	A	A	A	A	VOR/DME	CLIMB	{5,6}		¢
	A	A	A	c	AIR DATA	CRUISE I	{5,6}		¢
	A	A	C	*	INERTIAL	CRUISE II	{5,6}		¢
	¢	¢	A	*	AUTOLAND	CRUISE III	{5,6}		1
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{2',3}		¢
a(2)	*	*	*	*	AIDS	TAKEOFF	{5,6}	x	¢
	A	A	A	A	VOR/DME	CLIMB	{5,6}		¢
	A	A	A	d	AIR DATA	CRUISE I	{5,6}		¢
	A	A	A	*	INERTIAL	CRUISE II	{5,6}		¢
	¢	¢	A	*	AUTOLAND	CRUISE III	{5,6}		1
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	3		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{2',3,4,5,6}		¢
a(2)	*	*	*	*	AIDS	TAKEOFF	{5,6}	x	¢
	A	A	A	A	VOR/DME	CLIMB	{5,6}		¢
	A	A	A	c	AIR DATA	CRUISE I	{5,6}		¢
	A	A	b	*	INERTIAL	CRUISE II	{5,6}		¢
	¢	¢	A	*	AUTOLAND	CRUISE III	{2',3,4}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4,5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{2',3}		¢
a(2)	*	*	*	*	AIDS	TAKEOFF	{5,6}	x	¢
	A	A	A	A	VOR/DME	CLIMB	{5,6}		¢
	A	A	A	d	AIR DATA	CRUISE I	{5,6}		¢
	A	A	b	*	INERTIAL	CRUISE II	{5,6}		¢
	¢	¢	A	*	AUTOLAND	CRUISE III	{2',3,4}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4,5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	3		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{2',3,4,5,6}		¢

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise b
Column 3: Phase 3 = Cruise c
Column 4: Phase 4 = Landing

Names are defined in Table 14.

For each row, the resulting level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 2 BASED
CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 2 TRAJECTCRY SET NAMES FOR LEVEL 1 TRAJECTORIES					RESULTING LEVEL 2 TRAJECTORY SETS			
a(2)	*	*	*	*	AIDS	TAKEOFF	{5,6}		∅
	A	A	A	A	VOR/DME	CLIMB	{5,6}		∅
	A	A	A	e	AIR DATA	CRUISE I	{5,6}		∅
	A	A	b	*	INERTIAL	CRUISE II	{5,6}	x	∅
	∅	∅	A	*	AUTOLAND	CRUISE III	{2',3,4}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4,5,6}		∅
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		∅
	A	A	A	A	ATTITUDE CONTROL	LANDING	{3,4,5,6}		∅
a(2)	*	*	*	*	AIDS	TAKEOFF	{5,6}		∅
	A	A	A	A	VOR/DME	CLIMB	{5,6}		∅
	A	A	A	c	AIR DATA	CRUISE I	{5,6}		∅
	A	A	d	*	INERTIAL	CRUISE II	{5,6}	x	∅
	∅	∅	A	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4}		∅
	A	A	A	A	ENGINE CONTROL	APPROACH	{4,5,6}		∅
	A	A	A	A	ATTITUDE CONTROL	LANDING	{2',3}		∅
a(2)	*	*	*	*	AIDS	TAKEOFF	{5,6}		∅
	A	A	A	A	VOR/DME	CLIMB	{5,6}		∅
	A	A	A	d	AIR DATA	CRUISE I	{5,6}		∅
	A	A	d	*	INERTIAL	CRUISE II	{5,6}	x	∅
	∅	∅	A	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4}		∅
	A	A	A	A	ENGINE CONTROL	APPROACH	3		∅
	A	A	A	A	ATTITUDE CONTROL	LANDING	{2',3,4,5,6}		∅
a(2)	*	*	*	*	AIDS	TAKEOFF	{5,6}		∅
	A	A	A	A	VOR/DME	CLIMB	{5,6}		∅
	A	A	A	e	AIR DATA	CRUISE I	{5,6}		∅
	A	A	d	*	INERTIAL	CRUISE II	{5,6}	x	∅
	∅	∅	A	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4}		∅
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		∅
	A	A	A	A	ATTITUDE CONTROL	LANDING	{3,4,5,6}		∅
a(2)	*	*	*	*	AIDS	TAKEOFF	{5,6}		∅
	A	A	A	A	VOR/DME	CLIMB	{5,6}		∅
	A	A	A	e	AIR DATA	CRUISE I	{5,6}		∅
	A	A	e	*	INERTIAL	CRUISE II	{5,6}	x	∅
	∅	∅	A	*	AUTOLAND	CRUISE III	{5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		∅
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		∅
	A	A	A	A	ATTITUDE CONTROL	LANDING	{3,4,5,6}		∅

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise b
Column 3: Phase 3 = Cruise c
Column 4: Phase 4 = Landing

Names are defined in Table 14.

For each row, the resulting level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 2 BASED
CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 2 TRAJECTORY SET NAMES FOR LEVEL 1 TRAJECTORIES					RESULTING LEVEL 2 TRAJECTORY SETS			
	*	*	*	*	AIDS	TAKEOFF	{2,3,4}		¢
	*	*	*	*	VOR/DME	CLIMB	{2',3,4,5,6}		¢
	*	*	*	*	AIR DATA	CRUISE I	{2',3,4,5,6}		¢
a(3)	b	*	*	*	INERTIAL	CRUISE II	*	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	*		*
	A	*	*	*	ACTIVE FLUTTER CONTROL	DESCENT	*		¢
	A	*	*	*	ENGINE CONTROL	APPROACH	*		¢
	A	*	*	*	ATTITUDE CONTROL	LANDING	*		¢
	*	*	*	*	AIDS	TAKEOFF	{5,6}		¢
	*	*	*	*	VOR/DME	CLIMB	{2',3,4}		¢
	*	*	*	*	AIR DATA	CRUISE I	{2',3,4,5,6}		¢
	c	*	*	*	INERTIAL	CRUISE II	*	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	*		*
	A	*	*	*	ACTIVE FLUTTER CONTROL	DESCENT	*		¢
	A	*	*	*	ENGINE CONTROL	APPROACH	*		¢
	A	*	*	*	ATTITUDE CONTROL	LANDING	*		¢
	*	*	*	*	AIDS	TAKEOFF	{5,6}		¢
	*	*	*	*	VOR/DME	CLIMB	{5,6}		¢
	*	*	*	*	AIR DATA	CRUISE I	{2',3,4}		¢
a(3)	d	*	*	*	INERTIAL	CRUISE II	*	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	*		*
	A	*	*	*	ACTIVE FLUTTER CONTROL	DESCENT	*		¢
	A	*	*	*	ENGINE CONTROL	APPROACH	*		¢
	A	*	*	*	ATTITUDE CONTROL	LANDING	*		¢
	*	*	*	*	AIDS	TAKEOFF	{5,6}		¢
	*	*	*	*	VOR/DME	CLIMB	{5,6}		¢
	*	*	*	*	AIR DATA	CRUISE I	{5,6}		¢
a(3)	A	b	*	*	INERTIAL	CRUISE II	{2',3,4}	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	{2',3,4,5,6}		0
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4,5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{2',3,4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{2',3,4,5,6}		¢
	*	*	*	*	AIDS	TAKEOFF	{5,6}		¢
	*	*	*	*	VOR/DME	CLIMB	{5,6}		¢
	*	*	*	*	AIR DATA	CRUISE I	{5,6}		¢
a(3)	A	b	b	*	INERTIAL	CRUISE II	{2',3,4}	x	¢
	¢	¢	A	*	AUTOLAND	CRUISE III	{2',3,4,5,6}		1
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4,5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{2',3,4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{2',3,4,5,6}		¢

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise b
Column 3: Phase 3 = Cruise c
Column 4: Phase 4 = Landing

Names are defined in Table 14.

For each row, the resulting level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 2 BASED
CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 2 TRAJECTORY SET NAMES FOR LEVEL 1 TRAJECTORIES					RESULTING LEVEL 2 TRAJECTORY SETS			
a(3)	*	*	*	*	AIDS	TAKEOFF	{5,6}	x	¢
	*	*	*	*	VOR/DME	CLIMB	{5,6}		¢
	*	*	*	*	AIR DATA	CRUISE I	{5,6}		¢
	A	b	c	*	INERTIAL	CRUISE II	{2',3,4}		¢
	¢	¢	A	*	AUTOLAND	CRUISE III	{5,6}		1
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4,5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{2',3,4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{2',3,4,5,6}		¢
a(3)	*	*	*	*	AIDS	TAKEOFF	{5,6}	x	¢
	*	*	A	*	VOR/DME	CLIMB	{5,6}		¢
	*	*	A	*	AIR DATA	CRUISE I	{5,6}		¢
	A	A	b	*	INERTIAL	CRUISE II	{5,6}		¢
	¢	¢	A	*	AUTOLAND	CRUISE III	{2',3,4}		1
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4,5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{2',3,4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{2',3,4,5,6}		¢
a(3)	*	*	*	*	AIDS	TAKEOFF	{5,6}	x	¢
	*	*	A	*	VOR/DME	CLIMB	{5,6}		¢
	*	*	A	*	AIR DATA	CRUISE I	{5,6}		¢
	A	A	d	*	INERTIAL	CRUISE II	{5,6}		¢
	¢	¢	A	*	AUTOLAND	CRUISE III	{5,6}		1
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	{2',3,4,5,6}		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{2',3,4,5,6}		¢
a(3)	*	*	*	*	AIDS	TAKEOFF	{5,6}	x	¢
	*	*	A	*	VOR/DME	CLIMB	{5,6}		¢
	*	*	A	*	AIR DATA	CRUISE I	{5,6}		¢
	A	A	e	*	INERTIAL	CRUISE II	{5,6}		¢
	¢	¢	A	*	AUTOLAND	CRUISE III	{5,6}		1
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{5,6}		¢
	A	A	A	A	ENGINE CONTROL	APPROACH	2'		¢
	A	A	A	A	ATTITUDE CONTROL	LANDING	{2',3,4,5,6}		¢

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise b
Column 3: Phase 3 = Cruise c
Column 4: Phase 4 = Landing

Names are defined in Table 14.

For each row, the resulting level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 2 BASED CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 2 TRAJECTCRY SET NAMES FOR LEVEL 1 TRAJECTORIES					RESULTING LEVEL 2 TRAJECTORY SETS			
	*	*	*	*	AIDS	TAKEOFF	1		¢
	*	*	*	*	VOR/DME	CLIMB	*		¢
	*	*	*	*	AIR DATA	CRUISE I	*		¢
a(4)	*	*	*	*	INERTIAL	CRUISE II	*	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	*		*
	b	*	*	*	ACTIVE FLUTTER CONTROL	DESCENT	*		¢
	*	*	*	*	ENGINE CONTROL	APPROACH	*		¢
	*	*	*	*	ATTITUDE CONTROL	LANDING	*		¢
	*	*	*	*	AIDS	TAKEOFF	{2,3,4,5,6}		¢
	*	*	*	*	VOR/DME	CLIMB	1		¢
	*	*	*	*	AIR DATA	CRUISE I	*		¢
a(4)	*	*	*	*	INERTIAL	CRUISE II	*	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	*		*
	c	*	*	*	ACTIVE FLUTTER CONTROL	DESCENT	*		¢
	*	*	*	*	ENGINE CONTROL	APPROACH	*		¢
	*	*	*	*	ATTITUDE CONTROL	LANDING	*		¢
	*	*	*	*	AIDS	TAKEOFF	{2,3,4,5,6}		¢
	*	*	*	*	VOR/DME	CLIMB	{2,2',3,4,5,6}		¢
	*	*	*	*	AIR DATA	CRUISE I	1		¢
a(4)	*	*	*	*	INERTIAL	CRUISE II	*	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	*		*
	d	*	*	*	ACTIVE FLUTTER CONTROL	DESCENT	*		¢
	*	*	*	*	ENGINE CONTROL	APPROACH	*		¢
	*	*	*	*	ATTITUDE CONTROL	LANDING	*		¢
	*	*	*	*	AIDS	TAKEOFF	{2,3,4,5,6}		¢
	*	*	*	*	VOR/DME	CLIMB	2		¢
	*	*	*	*	AIR DATA	CRUISE I	{2,2',3,4,5,6}		¢
a(4)	*	*	*	*	INERTIAL	CRUISE II	*	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	*		*
	A	*	*	*	ACTIVE FLUTTER CONTROL	DESCENT	*		¢
	c	*	*	*	ENGINE CONTROL	APPROACH	*		¢
	*	*	*	*	ATTITUDE CONTROL	LANDING	*		¢
	*	*	*	*	AIDS	TAKEOFF	{2,3,4,5,6}		¢
	*	*	*	*	VOR/DME	CLIMB	{2',3,4,5,6}		¢
	*	*	*	*	AIR DATA	CRUISE I	2		¢
a(4)	*	*	*	*	INERTIAL	CRUISE II	*	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	*		*
	A	*	*	*	ACTIVE FLUTTER CONTROL	DESCENT	*		¢
	d	*	*	*	ENGINE CONTROL	APPROACH	*		¢
	*	*	*	*	ATTITUDE CONTROL	LANDING	*		¢

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise b
Column 3: Phase 3 = Cruise c
Column 4: Phase 4 = Landing

Names are defined in Table 14.

For each row, the resulting level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 2 BASED
CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 2 TRAJECTORY SET NAMES FOR LEVEL 1 TRAJECTORIES					RESULTING LEVEL 2 TRAJECTORY SETS			
	*	*	*	*	AIDS	TAKEOFF	{5,6}		ε
	*	*	*	*	VOR/DME	CLIMB	{5,6}		ε
	*	*	*	*	AIR DATA	CRUISE I	{5,6}		ε
a(4)	A	*	*	*	INERTIAL	CRUISE II	[illegible]	x	ε
	ε	ε	*	*	AUTOLAND	CRUISE III	*		*
	A	b	*	*	ACTIVE FLUTTER CONTROL	DESCENT	*		ε
	A	*	*	*	ENGINE CONTROL	APPROACH	*		ε
	A	*	*	*	ATTITUDE CONTROL	LANDING	*		ε
	*	*	*	*	AIDS	TAKEOFF	{5,6}		ε
	*	*	*	*	VOR/DME	CLIMB	{5,6}		ε
	*	*	*	*	AIR DATA	CRUISE I	{5,6}		ε
a(4)	A	*	*	*	INERTIAL	CRUISE II	2	x	ε
	ε	ε	*	*	AUTOLAND	CRUISE III	*		*
	A	A	*	*	ACTIVE FLUTTER CONTROL	DESCENT	*		ε
	A	b	*	*	ENGINE CONTROL	APPROACH	*		ε
	A	*	*	*	ATTITUDE CONTROL	LANDING	*		ε
	*	*	*	*	AIDS	TAKEOFF	{5,6}		ε
	*	*	*	*	VOR/DME	CLIMB	{5,6}		ε
	*	*	*	*	AIR DATA	CRUISE I	{5,6}		ε
a(4)	A	*	*	*	INERTIAL	CRUISE II	{2',3,4,5,6}	x	ε
	ε	ε	*	*	AUTOLAND	CRUISE III	1		*
	A	A	b	*	ACTIVE FLUTTER CONTROL	DESCENT	*		ε
	A	A	*	*	ENGINE CONTROL	APPROACH	*		ε
	A	A	*	*	ATTITUDE CONTROL	LANDING	*		ε
	*	*	*	*	AIDS	TAKEOFF	{5,6}		ε
	*	*	*	*	VOR/DME	CLIMB	{5,6}		ε
	*	*	*	*	AIR DATA	CRUISE I	{5,6}		ε
a(4)	A	*	*	*	INERTIAL	CRUISE II	{2',3,4,5,6}	x	ε
	ε	ε	*	*	AUTOLAND	CRUISE III	{2,2',3,4,5,6}		*
	A	A	c	*	ACTIVE FLUTTER CONTROL	DESCENT	1		ε
	A	A	*	*	ENGINE CONTROL	APPROACH	*		ε
	A	A	*	*	ATTITUDE CONTROL	LANDING	*		ε
	*	*	*	*	AIDS	TAKEOFF	{5,6}		ε
	*	*	*	*	VOR/DME	CLIMB	{5,6}		ε
	*	*	*	*	AIR DATA	CRUISE I	{5,6}		ε
a(4)	A	*	*	*	INERTIAL	CRUISE II	{2',3,4,5,6}	x	ε
	ε	ε	*	*	AUTOLAND	CRUISE III	{2,2',3,4,5,6}		*
	A	A	d	*	ACTIVE FLUTTER CONTROL	DESCENT	{2,2',3,4,5,6}		ε
	A	A	*	*	ENGINE CONTROL	APPROACH	1		ε
	A	A	*	*	ATTITUDE CONTROL	LANDING	*		ε

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise b
Column 3: Phase 3 = Cruise c
Column 4: Phase 4 = Landing

Names are defined in Table 14.

For each row, the resulting level 1 trajectory set is the intersection of the sets named in Column 2.

LEVEL 2 BASED CAPABILITY FUNCTION

ACCOMPLISHMENT LEVEL	LEVEL 2 TRAJECTORY SET NAMES FOR LEVEL 1 TRAJECTORIES					RESULTING LEVEL 2 TRAJECTORY SETS			
	*	*	*	*	AIDS	TAKEOFF	{5,6}		¢
	*	*	*	*	VOR/DME	CLIMB	{5,6}		¢
	*	*	*	*	AIR DATA	CRUISE I	{5,6}		¢
a(4)	A	*	*	*	INERTIAL	CRUISE II	{2',3,4,5,6}	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	2		*
	A	A	A	*	ACTIVE FLUTTER CONTROL	DESCENT	{2,2',3,4,5,6}		¢
	A	A	b	*	ENGINE CONTROL	APPROACH	{2,2',3,4,5,6}		¢
	A	A	*	*	ATTITUDE CONTROL	LANDING	*		¢
	*	*	*	*	AIDS	TAKEOFF	{5,6}		¢
	*	*	*	*	VOR/DME	CLIMB	{5,6}		¢
	*	*	*	*	AIR DATA	CRUISE I	{5,6}		¢
a(4)	A	*	*	*	INERTIAL	CRUISE II	{2',3,4,5,6}	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	{2',3,4,5,6}		*
	A	A	D	*	ACTIVE FLUTTER CONTROL	DESCENT	2		¢
	A	A	c	*	ENGINE CONTROL	APPROACH	{2,2',3,4,5,6}		¢
	A	A	*	*	ATTITUDS CONTROL	LANDING	*		¢
	*	*	*	*	AIDS	TAKEOFF	{5,6}		¢
	*	*	*	*	VOR/DME	CLIMB	{5,6}		¢
	*	*	*	*	AIR DATA	CRUISE I	{5,6}		¢
a(4)	A	*	*	*	INERTIAL	CRUISE II	{2',3,4,5,6}	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	{2',3,4,5,6}		*
	A	A	B	*	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4,5,6}		¢
	A	A	d	*	ENGINE CONTROL	APPROACH	2		¢
	A	A	*	*	ATTITUDE CONTROL	LANDING	*		¢
	*	*	*	*	AIDS	TAKEOFF	{5,6}		¢
	*	*	*	*	VOR/DME	CLIMB	{5,6}		¢
	*	*	*	*	AIR DATA	CRUISE I	{5,6}		¢
a(4)	A	*	*	*	INERTIAL	CRUISE II	{2',3,4,5,6}	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	{2',3,4,5,6}		*
	A	A	A	b	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4,5,6}		¢
	A	A	A	*	ENGINE CONTROL	APPROACH	{2',3,4,5,6}		¢
	A	A	A	*	ATTITUDE CONTROL	LANDING	1		¢
	*	*	*	*	AIDS	TAKEOFF	{5,6}		¢
	*	*	*	*	VOR/DME	CLIMB	{5,6}		¢
	*	*	*	*	AIR DATA	CRUISE I	{5,6}		¢
a(4)	A	*	*	*	INERTIAL	CRUISE II	{2',3,4,5,6}	x	¢
	¢	¢	*	*	AUTOLAND	CRUISE III	{2',3,4,5,6}		*
	A	A	A	A	ACTIVE FLUTTER CONTROL	DESCENT	{2',3,4,5,6}		¢
	A	A	A	b	ENGINE CONTROL	APPROACH	{2',3,4,5,6}		¢
	A	A	A	*	ATTITUDE CONTROL	LANDING	2		¢

Column 1: Phase 1 = Takeoff/Cruise A
Column 2: Phase 2 = Cruise b
Column 3: Phase 3 = Cruise c
Column 4: Phase 4 = Landing

Names are defined in Table 14.

For each row, the resulting level 1 trajectory set is the intersection of the sets named in Column 2.

earlier, can be ignored), $\xi_{ij}(U_1) = 0$. The level 2 Cartesian trajectory sets corresponding to these coordinate values are given by Table 14, e.g., $\xi_{11}(U_1) = 0$ says AIDS(1) = 0 which, by page 1, row 1 of the table, corresponds to the level 2 trajectory set

[6 6 6 * * * * *].

(There is only one Cartesian component in this case. In general these will be several, e.g., AIDS(1) = 1 yields the three components of page 1, rows 2-4).

Repeating this step for the remaining coordinate pairs (i,j) and using the names indicated in the last column of Table 14, the intersection of equation (3.3.2) is performed symbolically (as illustrated in the example of Section 3.3.1.3). Each product term of this expression is displayed in matrix form in the second column of Table 15; in this case there happens to be only one product term, which appears on page 1, row 1, column 2. The matrix arrangement of the names resolves their ambiguities, i.e., the A that appears as entry (i,j) names a Cartesian component that maps into $\xi_{ij}(U_1)$. Column 3 of Table 3 gives the resulting intersection of level 2 Cartesian sets named by entries in the matrix, along with the weather trajectories V_1 that are carried down from level 1. (Note that the matrix representation in column 3 has a different orientation, due to space limitations.) Thus, for the case in point, the computation yields the trajectory set

$$\begin{array}{r} \text{TAKEOFF} \\ \text{CLIMB} \\ \text{CRUISE I} \\ \text{CRUISE II} \\ \text{CRUISE III} \\ \text{DESCENT} \\ \text{APPROACH} \\ \text{LANDING} \end{array} \begin{bmatrix} 6 \\ 6 \\ 6 \\ 6 \\ 6 \\ 6 \\ \{5,6\} \\ \{5,6\} \end{bmatrix} \times \begin{bmatrix} \emptyset \\ \emptyset \\ \emptyset \\ \emptyset \\ 0 \\ \emptyset \\ \emptyset \\ \emptyset \end{bmatrix}$$

These computations are then repeated for U_2, U_3, and U_4. (Note that, unlike Table 3, computations that result in null sets are omitted from the tabluation.) Carrying out these computations, we find that there is only one other distinct Cartesian component associated with a_0 (displayed in Table 15, page 1, row 2, column 3) and hence

$$\gamma^{-1}(a_0) = \begin{array}{r} \text{TAKEOFF} \\ \text{CLIMB} \\ \text{CRUISE I} \\ \text{CRUISE II} \\ \text{CRUISE III} \\ \text{DESCENT} \\ \text{APPROACH} \\ \text{LANDING} \end{array} \begin{bmatrix} 6 \\ 6 \\ 6 \\ 6 \\ 6 \\ 6 \\ \{5,6\} \\ \{5,6\} \end{bmatrix} \times \begin{bmatrix} \emptyset \\ \emptyset \\ \emptyset \\ \emptyset \\ 0 \\ \emptyset \\ \emptyset \\ \emptyset \end{bmatrix}$$

$$\cup \quad \begin{array}{r} \text{TAKEOFF} \\ \text{CLIMB} \\ \text{CRUISE I} \\ \text{CRUISE II} \\ \text{CRUISE III} \\ \text{DESCENT} \\ \text{APPROACH} \\ \text{LANDING} \end{array} \begin{bmatrix} 6 \\ 6 \\ 6 \\ 6 \\ 6 \\ 6 \\ \{5,6\} \\ \{5,6\} \end{bmatrix} \times \begin{bmatrix} \emptyset \\ \emptyset \\ \emptyset \\ \emptyset \\ 1 \\ \emptyset \\ \emptyset \\ \emptyset \end{bmatrix}$$

This concludes the example.

On examining Table 3, one can observe that the remaining levels (a_1, a_2, a_3, a_4) involve more complex trajectory sets that are more difficult to determine. This is something we have observed before in earlier experiments, namely, that the base model trajectory set associated with the "most successful" level of performance tends to be "quite" Cartesian (i.e., has few Cartesian components in its decomposition), while those of lower levels do not. Indeed, the level a_0 trajectory set computed in the above example is purely Cartesian since the two components produced by the computational procedure differ only in the value of the phase 5 weather variable. Hence, relative to accomplishment level a_0, the phases of the base model are independent (see Section 3.1.3, corollary to Theorem 5). This says in turn that if level a_0 were taken to be "success" and the remaining levels were regarded as "failure," the resulting capability function would be structure-based (see Section 3.1.4, Theorem 6). In other words, the evaluation of SIFT could have been based on more conventional reliability models if "top performance" were the only concern. However, an examination of Table 15 reveals that the trajectory sets $\gamma^{-1}(a)$ for levels a_1, a_2, a_3 and a_4 are "far" from being Cartesian and, accordingly, there exist interphase dependencies relative to these lower levels of accomplishment. It is at these lower levels, then, that the full generality of our evaluation techniques must be brought into play.

3.3.4 Derivation of Transition Probabilities

Given the transition graphs of Figure 3 and Figure 4, there is enough information to determine the initial to final state transition matrix P(k), for each phase k ($k=1,2,\ldots,8$). There are several standard techniques for obtaining the initial to final state matrices (see [10], for example). However, for this particular SIFT model, these matrices can be obtained more easily using combinational probability methods (see Section 3.2.2). This is due to the assumption that each unit fails independently with a constant failure rate.

For the first phase, the initial to final state transition matrix is a $nm-(n+m)+2$ by $nm-(n+m)+2$ matrix

$$P(1) = [p_{T_1}[(i,j),\ (i',j')]]$$

where

$nm-(n+m)+2$ = the number of states of the phase 1 model (see Figure 6)

and

$p_{T_1}[(i,j),(i',j')]$ = the probability that the phase one model is in state (i',j') at time $t_1 = t_0 + T_1$ given that the phase one model is in state (i,j) at time t_0

= Prob[i' processor-memory units remain at time t_1 | i processor-memory units are available at time t_0] · Prob[j' busses remain at time t_1 | j busses are available at time t_0]

$$= \begin{cases} \binom{i}{i'} e^{-i'pt}(1-e^{-pt})^{i-i'} \cdot \binom{j}{j'} e^{-j'qt}(1-e^{-qt})^{j-j'} \\ \text{if } 2 \le i' \le i \le n \text{ and } 2 \le j' \le j \le m; \\ 0 \text{ otherwise.} \end{cases}$$

Finally, the probability of being in state F at time t_1 given that the phase 1 model is in state (i,j) at time t_0 can be expressed as

$$[\text{illegible}]\ j),F] = 1 - \sum_{j'=2}^{j} \sum_{i'=2}^{i} p_{T_1}[(i,j),(i',j')].$$

The initial to final state transition matrix for the second phase is an nm-n+1 by nm-n+1 matrix

$$P(2) = p_{T_2}[(i,j),\ (i',\ j')]$$

where

nm-n+1 = the number of states of the second phase model (see Figure 7)

$T_2 = t_2 - t_1$ = the duration of phase 2

$$p_{T_2}[(i,j),(i',j')] = \begin{cases} \binom{i}{i'} e^{-i'pT_2}(1-e^{-pT_2})^{i-i'} \cdot \binom{j}{j'} e^{-j'qT_2}(1-e^{-qT_2})^{j-j'} \\ \qquad \text{if } 3 \le i' \le i \le n \text{ and } 2 \le j' \le j \le m, \\ 0 \text{ if } 3 \le i < i' \le n \text{ or } 2 \le j < j' \le m. \end{cases}$$

When 2 processors and j' busses are available at the end of phase 2, two states (2, j') and (2', j') are created to distinguish two possible configurations. Corresponding with these two states, the transition probabilities are expressed as

$$p_{T_2}[(i,j),(2,j')] = \begin{cases} \binom{j}{j'} e^{-j'qT_2}(1-e^{-qT_2})^{j-j'} \cdot 2\binom{i}{3}(1-e^{-pT_2})^{i-2} e^{-2pT_2} \\ \qquad \text{if } 3 \le i \le n \text{ and } 2 \le j' \le j \le m, \\ \binom{j}{j'} e^{-j'qT_2}(1-e^{-qT_2})^{j-j'} \cdot e^{-2pT_2} \\ \qquad \text{if } i = 2 \text{ and } 2 \le j' \le j \le m, \\ 0 \text{ otherwise,} \end{cases}$$

$$p_{T_2}[(i,j),(2',j')] = \begin{cases} \binom{j}{j'} e^{-j'qT_2}(1-e^{-qT_2})^{j-j'} \cdot \binom{i}{3}(1-e^{-pT_2})^{i-2} e^{-2pT_2} \\ \qquad \text{if } 3 \le i \le n \text{ and } 2 \le j' \le j \le m, \\ 0 \text{ otherwise,} \end{cases}$$

and $p_{T_2}[(2',j),\ (2,j')] = 0$ for $2 \le j' \le j \le m$.

Furthermore,

$$p_{T_2}[(2',j),\ (2',j')] = \begin{cases} \binom{j}{j'} e^{-j'qT_2}(1-e^{-qT_2})^{j-j'} \cdot e^{-2pT_2} \\ \qquad \text{if } 2 \le j' \le j \le m, \\ 0 \text{ otherwise.} \end{cases}$$

Finally,

$$p_{T_2}[(i,j),F] = 1 - \sum_{j'=2}^{j} \sum_{i'=2}^{i} p_{T_2}[(i,j),(i',j')] - \sum_{j'=2}^{j} p_{T_2}[(i,j),(2',j')]$$

Since the remaining phases have the same transition graph as the second phase, the corresponding initial to final phase transition matrices can be represented as nm-n+1 by nm-n+1 matrices

$$P(k) = [p_{T_k}[(i,j),(i',j')]]$$

where for each k = 3, 4, ..., 8 T_k is the duration of the kth phase and $p_{T_k}[(i,j),(i',j')]$ is defined as above with T_2 replaced by T_k.

Since the underlying Markov models differ for different phases, it is also necessary to specify the interphase transition matrices (see [3], Section 3.4.3). Generally, the interphase transition matrix H(k) is defined to be an n_k by n_{k+1} matrix

$$H(k) = [h_{ij}]$$

where n_k and n_{k+1} are the number of states for the k^{th} and the $k+1^{th}$ phase models and

h_{ij} = probability that the state of the phase k+1 model is j (at time t_k) given that the state of the phase k model is i (at time t_k).

For the phase models described above, the phase 1 model has $n_1 = nm-(n+m)+2$ states, the phase 2 model has $n_2 = nm-n+1$ states and the interphase transition matrix $H(1)$ is the following n_1 by n_2 matrix.

$$\begin{array}{c|ccccccccccc|}
 & (n,m) & (n,m-1) & \ldots & (2,m) & (2',m) & (2,m-1) & (2',m-1) & \ldots & (2,2) & (2',2) & F \\
\hline
(n,m) & 1 & & & & & & & & & & \\
(n,m-1) & & 1 & & & & & \bigcirc & & & & \\
\vdots & & & \ddots & & & & & & & & \\
(2,m) & & & & 1 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
(2,m-1) & & & & 0 & 0 & 1 & 0 & \ldots & 0 & 0 & 0 \\
\vdots & & \bigcirc & & & & \ddots & & & & & \\
(2,2) & & & & 0 & 0 & 0 & 0 & \ldots & 1 & 0 & 0 \\
F & & & & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 1 \\
\hline
\end{array}$$

The above matrix is interpreted as follows. When the phase 1 model is in the state (i,j) such that $n \geq i \geq 2$ and $m \geq j \geq 2$ at the end of phase 1, the computer reconfigures with probability 1 to state (i,j) of the phase 2 model. For the second phase, the interphase transition matrix $H(2)$ is represented by a n_2 by n_3 matrix $(n_2 = n_3 = nm-n+1)$.

$$\begin{array}{c|ccccccccccc}
 & (n,m) & \cdots & (2,m) & (2',m) & (2,m-1) & (2',m-1) & \cdots & (2,2) & (2',2) & F \\
\hline
(n,m) & 1 & & & & & \bigcirc & & & & \\
\vdots & & \ddots & & & & & & & & \\
(2,m) & & & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
(2',m) & & & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
(2,m-1) & \bigcirc & & 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\
(2',m-1) & & & 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\
\vdots & & & & & \ddots & & & & & \\
(2,2) & & & 0 & 0 & 0 & 0 & \cdots & 1 & 0 & 0 \\
(2',2) & & & 0 & 0 & 0 & 0 & \cdots & 1 & 0 & 0 \\
F & & & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1
\end{array}$$

As in the previous case, when the phase 2 model is in state (i,j), $n \geq i \geq 2$, $m \geq j \geq 2$, at the end of phase 2, the computer is assumed to have reconfigured into state (i,j) with probability 1. However, when the computer is in state (2',j) $m \geq j \geq 2$ at the end of phase 2, that is, the Engin Control has failed due to error latency, the computer is assumed to be capable of detecting this failure and reconfigures to state (2,j) of phase 3.

Applying the same argument to the remaining phases, the interphase transition matrices H(k), k = 3,4,5,6,7 are the same as H(2).

3.3.5 Performability Results

Having derived the trajectory sets associated with each accomplishment level in $\{a_0, a_1, \ldots, a_4\}$ (Section 3.3.3; Table 15) and the transition matrices of the computer model (Section 3.3.4), evaluation of a sepcific system requires designation of values for the following parameters of SHIFT and its environment:

COMPUTER (SIFT)

C1) Processor failure rate,

C2) Bus failure rate,

C3) Initial state distribution (i.e., availability of computer resources at takeoff).

ENVIRONMENT

E1) Flight duration and phase durations,

E2) Probability of Category III weather at destination airport.

Evaluations were performed for a number of specific systems determined by the following choices of parameter values.

C1) As in [8], the processor failure rate for each system is taken to be 10^{-4} failures per hour.

C2) As in [8], the bus failure rate for each system is taken to be 10^{-5} failures per hour.

C3) Two types of initial state distributions are considered. The first type is "deterministic" in the sense that one computer state has probability 1 of being the initial state (the remaining states having probability 0). If (i,j) is the state having probability 1 (recall that i is the number of fault-free processors; j is the number of fault-free busses), this distribution is denoted

$$\mathrm{Det}(i,j).$$

The second type of initial state distribution considered is truly probabilistic where one of two specific distributions are assumed. These are denoted I_1 and I_2 and are given by Table 16.

State	Distribution	
(i,j)	I_1	I_2
(6,6)	.64	.31
(6,5)	.128	.081
(6,4)	.032	.009
(5,6)	.16	.09
(5,5)	.032	.009
(5,4)	.008	.001
Others	0	0

TABLE 16

Initial State Probabilities

E1) Two filght missions are considered, a 6 hour and 25 minute flight from London to New York (JFK Airport) and a 10 hour flight from Tel Aviv to New York. The assumed phase durations associated with each flight are given in Table 17.

	Flight	
Phase	London-New York	Tel Aviv-New York
Takeoff	1 minute	1 minute
Climb	15 minutes	15 minutes
Cruise I	25 minutes	25 minutes
Cruise II	5 hours	8 hours 35 minutes
Cruise III	25 minutes	25 minutes
Descent	15 minutes	15 minutes
Approach	3 minutes	3 minutes
Landing	1 minute	1 minute
Total Duration	6 hours 25 minutes	10 hours

TABLE 17

Phase Durations

E2) The probability of Category III weather at JFK is taken to be .011 (see [11], p.173).

For the fixed values of C1, C2, and E2 indicated above and for choices of C3 and E1 as indicated in Table 18, 14 specific systems were evaluated (denoted $S_1, S_2, \ldots, S_{14}$). For each system S_i, the resulting performability p_{S_i} is also presented in Table 18, where the entry corresponding to system S_i and accomplishment level a_j is the probability $p_{S_i}(a_j)$.

To summarize the calculation of $p_{S_i}(a_j)$:

1) The base model trajectory set $U_{a_j} = \gamma_{S_i}^{-1}(a_j)$ is expressed as a union of Cartesian trajectory sets (see Table 15; these sets are common to each of the 14 specific systems).

2) The initial state distribution of S_i determines the initial state vector I(0). The flight of S_i, with its associated phase durations, determines the specific nature of the transition matrices P(1),..., P(8) and H(1),...,H(7) derived in Section 3.3.4.

3) For each Cartesian component V of U_{a_j}, V is represented by the characteristic matrices G(1),...,G(7) and F(8) (see [3], pp. 59-60) and Pr[V] is computed by the formula

$$\Pr[V] = I(0)\left(\prod_{m=1}^{7} P(m)G(m)H(m)\right)P(8)F(8).$$

(See [3], p.68, Theorem 3.)

4) The performability of S_i relative to level a_j is the sum, over all V in U_{a_j} of the Pr[V], i.e.,

$$p_{S_i}(a_j) = \sum_{V \subseteq U_{a_j}} \Pr[V].$$

Step 2) of the procedure outlined above was aided by METAPHOR wherein the DEDFAIL transition matrix generator function was employed to obtain the intraphase transition matrices P(m). The interphase transition matrices H(m) entered via the GIVEN command. In step 3), the matrices G(m)

Key:

	Economic Penalties	Operational Penalties	Change in Mission Profile	Fatalities
a_0	No	No	No	No
a_1	Yes	No	No	No
a_2	Maybe	Yes	No	No
a_3	Maybe	Maybe	Maybe	No
a_4	Maybe	Maybe	Maybe	Yes

System	C3 Init. State Distribution	E1 Flight	Accomplishment Level				
			a_0	a_1	a_2	a_3	a_4
S_1	Det(6,6)	Lon-NY	9.96×10^{-1}	3.80×10^{-3}	3.78×10^{-12}	6.02×10^{-6}	1.95×10^{-12}
S_2	Det(6,5)	Lon-NY	9.96×10^{-1}	3.80×10^{-3}	3.79×10^{-12}	6.02×10^{-6}	1.95×10^{-12}
S_3	Det(6,4)	Lon-NY	9.96×10^{-1}	3.80×10^{-3}	1.33×10^{-10}	6.05×10^{-6}	2.97×10^{-12}
S_4	Det(5,6)	Lon-NY	0	9.97×10^{-1}	1.03×10^{-9}	3.17×10^{-3}	1.55×10^{-9}
S_5	Det(5,5)	Lon-NY	0	9.97×10^{-1}	1.03×10^{-9}	3.17×10^{-3}	1.55×10^{-9}
S_6	Det(5,4)	Lon-NY	0	9.97×10^{-1}	1.16×10^{-9}	3.17×10^{-3}	1.55×10^{-9}
S_7	Det(6,6)	TA-NY	9.94×10^{-1}	6.03×10^{-3}	6.07×10^{-12}	1.52×10^{-5}	1.30×10^{-11}

TABLE 18

Performability Results

System	C3 Init. State Distribution	E1 Flight	Accomplishment Level				
			a_0	a_1	a_2	a_3	a_4
S_8	Det(6,5)	TA-NY	9.94×10^{-1}	6.03×10^{-3}	6.12×10^{-12}	1.52×10^{-5}	1.30×10^{-11}
S_9	Det(6,4)	TA-NY	9.94×10^{-1}	6.03×10^{-3}	2.09×10^{-10}	1.53×10^{-5}	1.71×10^{-11}
S_{10}	Det(5,6)	TA-NY	0	9.95×10^{-1}	1.03×10^{-9}	5.03×10^{-3}	7.15×10^{-9}
S_{11}	Det(5,5)	TA-NY	0	9.95×10^{-1}	1.03×10^{-9}	5.03×10^{-3}	7.15×10^{-9}
S_{12}	Det(5,4)	TA-NY	0	9.95×10^{-1}	1.23×10^{-9}	5.03×10^{-3}	7.15×10^{-9}
S_{13}	I_1	TA-NY	7.95×10^{-1}	2.04×10^{-1}	2.18×10^{-10}	1.02×10^{-3}	1.44×10^{-9}
S_{14}	I_2	TA-NY	8.95×10^{-1}	1.05×10^{-1}	1.10×10^{-10}	5.17×10^{-4}	7.26×10^{-10}

TABLE 18 - Continued
Performability Results

and F(8) were likewise entered via the GIVEN command. The calculations of step 3) and step 4) were executed by METAPHOR programs.

We make no attempt at this time to interpret the performability results of Table 18. The intent of this evaluation exercise was to further establish the practicality of performability evaluation as it applies to aircraft computing systems, and we believe this has been achieved by the effort reported herein. However, having developed this model hierarchy, we plan to obtain further evaluation data by choosing other values of parameters C1-C3 and E1-E2. This data, along with the performability results of Table 18, will then be examined for possible implications regarding the design and use of the SIFT computer.

4. REFERENCES

[1] "Models and Techniques for Evaluating the Effectiveness of Aircraft Computing Systems," Semi-Annual Status Report Number 1, NASA Grant NSG 1306, November 1976.

[2] "Models and Techniques for Evaluating the Effectiveness of Aircraft Computing Systems," Semi-Annual Status Report Number 2, NASA Grant NSG 1306, July 1977.

[3] "Models and Techniques for Evaluating the Effectiveness of Aircraft Computing Systems," Semi-Annual Status Report Number 3, NASA Grant NSG 1306, January 1978.

[4] J. F. Meyer, "On Evaluating the Performability of Degradable Computing Systems," Digest of the 8th Int'l Symp. on Fault-Tolerant Computing, Toulouse, France, pp. 44-49, June 1978.

[5] R. A. Ballance and J. F. Meyer, "Functional Dependence and Its Application to System Evaluation," Proc. of the 1978 Johns Hopkins Conference on Information Sciences and Systems, Baltimore, MD, pp. 280-285, March 1978.

[6] "Models and Techniques for Evaluating the Effectiveness of Aircraft Computing Systems," Proposal for extension on NASA Grant NSG 1306, submitted to NASA Langley Research Center, March 1977.

[7] R. S. Ratner, E. B. Shapiro, H. M. Zeidler, S. W. Wahlstrom, C. B. Clark and J. Goldberg, "Design of a Fault Tolerant Airborne Digital Computer," Vol. II, Stanford Research Institute, Final Report, NASA Contract NAS1-10920, October 1973.

[8] J. H. Wensley, J. Goldberg, M.W. Green, W. H. Kautz, K. N. Levitt, M. E. Mills, R. E. Shostak, P. Whiting-O'Keefe, and H. M. Zeidler, "Design Study of Software Implemented Fault Tolerance (SIFT) Computer, "Interim Stanford Research Institute, June 1978.

[9] J. D. Esary and H. Ziems, "Reliability of Phased Missions," Reliability and Fault Tree Analysis, SIAM, Philadelphia, PA, pp. 213-236, 1975.

[10] E. Cinlar, Introduction to Stochastic Processes, Prentice-Hall, Englewood Cliffs, NJ, 1975.